AD-A243 404

MENTATION PAGE

Form Approved
OMB No. 0704-0188

s estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, ng and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this ng this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1991	THESIS/~~DISSERTATION~~

4. TITLE AND SUBTITLE

A Convergent Aspiration Based Interior Point Method (CAIN) For Multiple Objective Linear Programming (MOLP)

5. FUNDING NUMBERS

6. AUTHOR(S)

Michael S. Mynes, Captain

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AFIT Student Attending: University of Oklahoma

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/CI/CIA- 91-086

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFIT/CI
Wright-Patterson AFB OH 45433-6583

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release IAW 190-1
Distributed Unlimited
ERNEST A. HAYGOOD, Captain, USAF
Executive Officer

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

14. SUBJECT TERMS

15. NUMBER OF PAGES

81

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

THE UNIVERSITY OF OKLAHOMA

A CONVERGENT ASPIRATION BASED INTERIOR POINT METHOD (CAIN)

FOR

MULTIPLE OBJECTIVE LINEAR PROGRAMMING (MOLP)

A Project Report Submitted to

the Graduate faculty in Partial

Fulfillment of the Degree of

MASTER OF SCIENCE

By M.S. MYNES

Norman, Oklahoma

August 1991

A CONVERGENT ASPIRATION BASED INTERIOR POINT METHOD

(CAIN)

FOR MULTIPLE OBJECTIVE LINEAR PROGRAMMING (MOLP)

A PROJECT APPROVED FOR

THE SCHOOL OF INDUSTRIAL ENGINEERING

BY

M.S. Comprehensive Exam
Summer 1991

Review the paper by Lotfi, Stewart, and Zionts:

"An Aspiration-level interactive model for Multiple Criteria Decision Making", working paper, 1990.

- Summarize the paper.
- Give a literature review.
- Extend this paper for the continuous case and try to apply it in the linear case.
- Prepare a report to document your results.

STUDENT: Mike Mynes

EXAMINING COMMITTEE: T. Trafalis (Chair)
S. Pulat
S. Raman

DATE GIVEN: May 17, 1991

DATE REPORT DUE: July 24, 1991

NOTES:

1. Contact the Chair for additonal information.
2. The report must follow the format of the M.S. Comprehensive Exam Report of Homer L. Pryor (available in the IE Library, Room 116). The report must have the question typed at the beginning.
3. Submit four copies of the final report - one to each Committee Member and one to the IE School.
4. Oral comprehensive exams will be held during the week following July 31, 1991. Contact Barbara Morris of the Graduate College and request "Authority to take the Comprehensive Exam" by July 24, 1991.

ABSTRACT

This report describes a new convergent aspiration based algorithm (CAIN - Convergent Aspiration based INterior method) for solving the multiple objective linear programming (MOLP) problem. Initial motivation for the research was provided by a recently developed methodology for the discrete multiple criteria decision making problem called AIM (Aspiration-Level Interactive model). Although CAIN uses many of the features implemented in AIM, the continuous MOLP provides for an entirely different domain of research. As part of CAIN, an innovative decision maker (DM) interaction technique called ALaRM (Aspiration Level Range Method) was concurrently developed. Using ALaRM, an interior point strategy for converging to efficient solutions is employed based upon DM levels of aspiration for the objectives. This technique, the Algorithm of Centers, has been shown to converge in polynomial time (unlike many simplex based strategies). CAIN is shown to be simple and practical from a DM standpoint, and is believed to represent an improvement over existing aspiration based MOLP techniques.

TABLE OF CONTENTS

LIST OF FIGURES

1.0 **INTRODUCTION**

Interactive techniques for solving the Multiple Criteria Decision Making (MCDM) problem refer to methods of obtaining (ideally, converging to) a nondominated (efficient, pareto-optimal) "best compromise solution" based upon progressive preference articulations of one or more decision maker(s) (DM). Drawing upon the ideas of various researchers, Lofti, Stewart, and Zionts (1990) have developed "An Aspiration Level Interactive Model (AIM) for MCDM". Motivated by the need for a simple yet practical interactive procedure, AIM solves the discrete alternative MCDM problem based upon the concept of DM levels of aspiration for a given set of objectives.

The primary objective of this research was to extend aspiration level concepts to the realm of continuous linear MCDM problems. Specifically, this paper shall present a conceptually simple and practical approach for solving multiple objective linear programming (MOLP) problems based upon DM levels of aspiration. The algorithm, hereafter known as CAIN (Convergent Aspiration-based INterior method) uses optimization concepts ranging from the most traditional to the most recent. In addition, an innovative method of interacting with a DM renders CAIN conceptually modest and user friendly.

Toward this end, section 2 begins with an examination of the generic MOLP problem formulation. After an understanding of MOLP concepts and terminology has been established, section 3 reviews some traditional MOLP solution techniques. Although every attempt

was made to be thorough, emphasis was placed on interactive algorithms as opposed to those requiring prior or "after the fact" articulation of DM desires. In addition, because of their impact on this research (indeed they provided the initial motivation), section 4 is devoted to detailed discussions of AIM, a continuous case variation of AIM called CASE (Convergent Aspiration level SEarch Method), and other more modern interactive algorithms developed for solving continuous decision problems. Subsequent to these reviews, the CAIN algorithm is developed in section 5. As implied by its nomenclature, CAIN uses an interior point technique as its method of convergence to a nondominated solution. While simple in concept, this method, known as the Algorithm of Centers (Trafalis, 1990) is mathematically advanced. Since it is not the intent of this report to overwhelm readers or potential users with mathematical theory, actual development of the Algorithm of Centers is left as Appendix A. Following formal development of CAIN, section 6 presents numerical examples featuring CAIN and a fictitious DM. Subjective comparisons between CAIN and other existing MOLP solution methodologies follow in section 7. Finally, section 8 includes conclusions and some recommendations for further research.

Throughout this paper, it is assumed that the reader is familiar with basic concepts of single objective optimization. More specifically, familiarity with such concepts as the simplex method, extreme points, extreme point adjacency, defining hyperplanes of a feasible region, feasibility, and pivoting is

necessary for full comprehension of traditional MOLP techniques. However, for the reader interested solely in CAIN and its development, only the most basic knowledge of the simplex algorithm is required for full conceptual comprehension.

2.0 MULTIPLE OBJECTIVE LINEAR PROGRAMMING (MOLP)

MOLP is a special class of the general MCDM problem in which all objective functions and constraints are linear. In general, formulations of MOLPs can exist in terms of decision variables (the decision space) or complete objective functions (the objective space). The following sections review both of these formulation strategies.

2.1 Formulation in the Decision Space

Given a set of p objective functions and m constraints, the general MOLP can be formulated in the decision space as:

$$
\begin{aligned}
&Max\ f_i(x) = c^{(i)}x && \forall i=1,2,\ldots,p && (1)\\
&s.t.\ \ a_j^t x \geq b_j && \forall j=1,2,\ldots,m && (2)\\
&\qquad x \geq 0
\end{aligned}
$$

where $c^{(i)} \equiv$ *1 x n vector of cost coefficients*
$x \equiv$ *n x 1 vector of decision variables*
$a_j \equiv$ *n x 1 vector of technological coefficients for the jth real constraint*
$b_j \equiv$ *right-hand-side constant for the jth real constraint*

Definition 2.1.1: The set defined by:

$$X = \left\{ x \,\middle|\, \begin{array}{ll} a_j^t x \geq b_j & \forall j=1,2,\ldots,m \\ x \geq 0 & \end{array} \right\}$$

is called the set of feasible decision vectors or the feasible region in the decision space.

Definition 2.1.2: A set X in E^n (E-space) is called a convex set if given any two points x_1 and x_2 in

X, $\lambda x_1 + (1-\lambda)x_2 \in X \quad \forall \lambda \in [0,1]$. *(Bazaraa, 1990)*

Definition 2.1.3: *A set X in E^n is* ***bounded*** *if there exists a number such that* $\|x\| < k \quad \forall x \in X$ *where* $\| \ \|$ *represents the Euclidean norm (Bazaraa, 1990).*

For purposes of this paper, X is assumed bounded and convex. Furthermore, the maximization convention is assumed for all objective functions. Recall that

$$Max\ c^{(i)}x = -\ Min\ -\ c^{(i)}x \quad \forall i=1,2,\ldots,p$$

Therefore, any minimization objective is easily converted to the maximization convention.

Unfortunately, most practical applications for MOLP involve objective functions which conflict and cannot all be maximized simultaneously. This dilemma is examined further in the context of "ideal" and "nadir" solutions in section 2.5.

2.2 Decision Space Example

A consumer wishes to purchase three products, A, B, and C. Each unit of product A costs \$4, each unit of product B costs \$3, and each unit of product C costs \$5. The consumer has only \$60 and two objectives: f_1: purchase the maximum number of units of products A and B combined, and f_2: purchase the maximum number of units of product C. Define the following variables:

$x_1 \equiv$ *number of units of product A to purchase*
$x_2 \equiv$ *number of units of product B to purchase*
$x_3 \equiv$ *number of units of product C to purchase*

The MOLP formulation in the decision space can then be expressed as follows.

$$\begin{aligned} &\mathit{Max}\ f_1 = x_1 + x_2 \\ &\mathit{Max}\ f_2 = x_3 \\ s.t.\ \ &4x_1 + 3x_2 + 5x_3 \le 60 \\ &x_1\ ,\ x_2\ ,\ x_3 \ge 0 \end{aligned}$$

$$\mathit{Here},\quad c^{(1)} \equiv (1,1,0),\ c^{(2)} \equiv (0,0,1)$$

$$x \equiv \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix},\ a_1 \equiv \begin{pmatrix} -4 \\ -3 \\ -5 \end{pmatrix}, \mathit{and}\ h_1 \equiv -60$$

Note the problem exists in E^3.

2.3 **Formulation in the Objective Space**

Assuming an initial MOLP formulation in terms of decision variables, X can be mapped into the set of feasible objective space vectors F giving MOLP formulation in the objective space. Algebraically, each decision variable (x_i, i=1,2,...n) is solved in terms of objective function variables (f_i, i=1,2,...,p). These expressions are then substituted for the decision variables in the defining equations of X. In set notation:

$$X \to F = \{ f \mid g_r^t f \ge h_r \quad \forall r=1,2,\ldots,m+n \} \qquad (3)$$

where $g_r \equiv$ *p x* 1 *vector of objective space constraint coefficients for the rth objective space constraint*
$f \equiv$ *p x* 1 *vector of objective function values for objectives* f_i, *i*=1,2,...,*p*
$h_r \equiv$ *right-hand-side element of the rth objective space constraint*

Definition 2.3.1: The set defined by:

$$F \equiv \{f \mid g_r^t f \geq h_r \quad \forall r=1,2,\ldots,m+n\}$$

is called the set of feasible objective vectors or the feasible region in the objective space.

2.4 Objective Space Example

For the example given in section 2.2, the objective space equivalent formulation can be written as:

$$\begin{aligned} & \text{Max } f_1 \\ & \text{Max } f_2 \\ \text{s.t. } & 3f_1 + 5f_2 \leq 60 \\ & f_1 \geq 0 \\ & f_2 \geq 0 \end{aligned}$$

$$\text{where } g_1 \equiv \begin{pmatrix}-3\\-5\end{pmatrix},\ g_2 \equiv \begin{pmatrix}1\\0\end{pmatrix},\ g_3 \equiv \begin{pmatrix}0\\1\end{pmatrix}$$

$$f \equiv \begin{pmatrix}f_1\\f_2\end{pmatrix}, \text{and } \begin{pmatrix}h_1\\h_2\\h_3\end{pmatrix} \equiv \begin{pmatrix}-60\\0\\0\end{pmatrix}$$

Note the formulation exists in E^2.

2.5 Observations

Notice that the objective space constraint coefficients (g_r, $r=1,2,\ldots,m+n$) and the right-hand-side elements (h_r, $r=1,2,\ldots,m+n$) can be algebraically adjusted to any convenient values maintaining consistency between X and F. Note also that in general, the existence of m+n objective space constraints is predicated by existence of the m real and n nonnegativity constraints of X. Often, as is the case with this example, some of the constraints prove redundant. Here, the nonnegativity constraints for both x_1 and x_2 map into the same constraint, $f_1 \geq 0$.

Two important observations should be recognized from the example mapping of X into F. First, the order of the problem was reduced. In the decision space, the problem formulation exists in E^3 whereas in the objective space, the order was reduced to E^2. This is often the case as there usually exist fewer objectives than decision variables ($p<n$). Secondly, note the simplicity of the objective functions in the objective space. Since each objective function is assigned a single objective space variable, the mapped objective functions will always require the maximization of a single unique variable. These advantages of objective space formulation have been integrated into the CAIN algorithm as shall become clear during its development in section 5. The advantages will be particularly evident during the DM interaction segment. <u>Because of these benefits, MOLP formulation hereafter is referred to in the context of the objective space</u>.

<u>*Definition 2.5.1*</u>: *An objective space vector f is termed an alternative iff $f \in F$.*

When considering possible alternatives to MOLP, a <u>rational</u> DM will only care to consider the set of <u>efficient</u> (nondominated, pareto-optimal) alternatives provided he or she is aware of their existence. Formally, an efficient alternative can be defined as follows.

<u>*Definition 2.5.2*</u>: *An alternative f^e is said to be an <u>efficient</u> (nondominated, pareto-optimal) solution of MOLP iff there exists no other*

alternative f such that:

$$f_i \geq f_i^e \quad \forall i=1,2,\ldots,p$$

An integral concept of the CAIN DM interaction phase is that of an ideal value.

<u>*Definition 2.5.3*</u>: *An <u>ideal</u> solution, f^* is a solution vector which simultaneously maximizes every objective f_i, $i=1,2,\ldots,p$ for a given MOLP.*

Clearly, if f^* is indeed feasible to MOLP, it would always be the most desirable solution. In practical problems, however, the objective functions conflict and the ideal solution is not achievable. Nevertheless, because the concept of the ideal is so key to the method of DM interaction used by CAIN, the reader should be familiar with its computation. For any objective, f_i, its ideal value can be found by maximizing f_i individually, ignoring all other objectives, subject to the defined feasible objective space. That is,

$$f_i^* = \begin{array}{l} \text{Max } f_i \\ \text{s.t. } \; f \in F \end{array} \quad \forall i=1,2,\ldots,p$$

The vector of all ideal values is f^*. A variation of this generic ideal definition used by CAIN is developed in section 5.

<u>*Definition 2.5.4*</u> *A <u>nadir</u> solution, f^N, is a solution vector which simultaneously minimizes every objective f_i, $i=1,2,\ldots,p$ for a given MOLP.*

Clearly, the nadir is the antithesis of the ideal and its

components can be computed for every f_i as

$$f_i^N = \begin{matrix} Min\ f_i \\ s.t.\quad f \in F \end{matrix} \quad \forall i=1,2,\ldots,p$$

As with the ideal, the vector of all nadir values is f^N.

2.6 Graphical Example

Fig. 2.1 shows the feasible objective space for the example

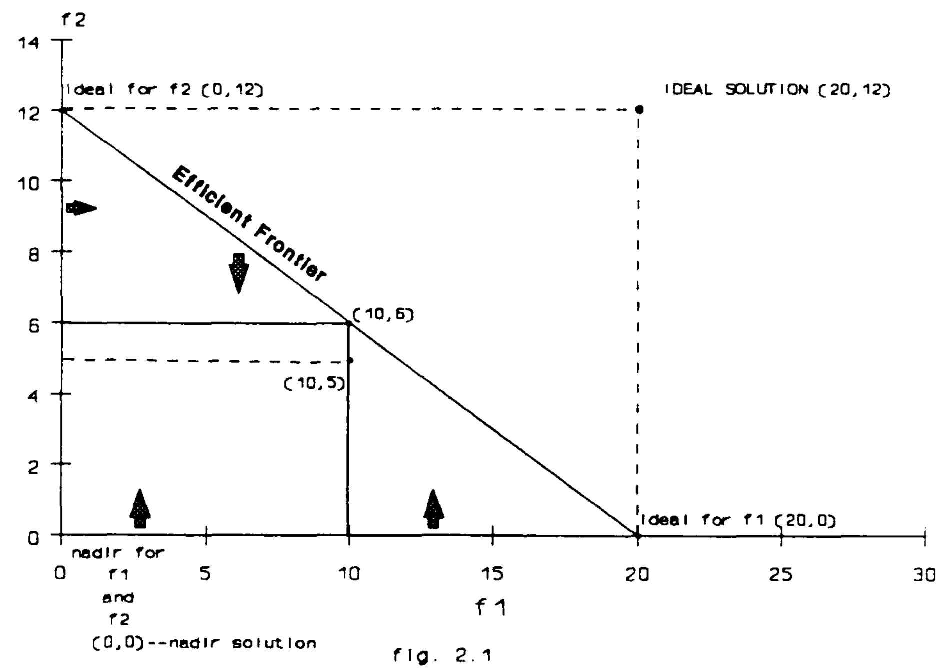


fig. 2.1

MOLP of section 2.4. Observe that point (10,5) is a dominated alternative since either of the objective functions (f_1 or f_2) could be increased without compromising the value of the other objective. For instance, alternative (10,6) is said to dominate alternative (10,5) since a gain is realized in f_2 without sacrificing on f_1. Furthermore, alternative (10,6) is an efficient alternative. Note that an increase in either objective at this point must be

accompanied by a decrease in the other objective in order for the alternative to remain feasible. Close examination of the feasible objective space F reveals that the set of all efficient alternatives lies along the defining hyperplane $3f_1 + 5f_2 = 60$. In the two dimensional objective space, the set of all efficient alternatives is always clearly visible via the "northeast rule" (under the maximization convention). Simply stated, the rule translates the axes of the positive quadrant to each point in F. If no other points feasible to F exist in the quadrant, then the alternative is efficient. This rule is not valid in higher dimensions, however, and identification of all efficient alternatives can be a difficult computational task.

<u>Definition 7</u>: *The set of all efficient alternatives, F^*, $F^* \in F$ lie along <u>efficient faces</u> (edges, surfaces, higher order hyperplanes) of F. The collection all efficient faces of F is termed the <u>efficient frontier</u>.*

As with the efficient vectors in the decision space X, the efficient frontier will always occur along the boundary of F (see Yu, 1985).

Finally, take note of the ideal and nadir solutions. As is usually the case for conflicting objectives, the ideal solution is infeasible. On the other hand, although not desired, the nadir solution for this example could be achieved.

Given this background, solution techniques for MOLP can be discussed. Section 3 explores some of the more popular traditional

techniques. Section 4 follows with a detailed review of the most recent and/or practical interactive algorithms for MOLP.

3.0 **TRADITIONAL MOLP SOLUTION METHODOLOGIES**

3.1 **Introduction**

The MOLP problem described in section 2 has been the motivation for development of numerous algorithms involving continuous decision variables. In order to better organize a review of such algorithms, a classification system based upon the timing of elicitation of preference information from a DM shall be followed. In the broadest sense, the timing of algorithm interaction with a DM has three possibilities. Some techniques employ what is known as prior articulation of DM preferences. In other words, complete interaction with the DM takes place before any algorithm computations begin. At the other extreme, techniques which employ posteriori articulation of preferences only confer with a DM after the particular algorithm has been completed. Between these extremes exist techniques which employ a progressive articulation of preferences. These algorithms gather information in stages (or iterations) coincident with the solution generating process. Ideally, the DM is guiding the process to his or her "best compromise solution". These algorithms are better known as "interactive" techniques and have been the subject of much recent research in the MCDM arena. A special subclass of the interactive techniques are the "aspiration-level" techniques, so called because of the type of information sought from the decision maker. CAIN falls into this class of interactive aspiration-level methods. Therefore, although this review shall bear mention of the entire spectrum of MOLP techniques, emphasis shall be placed upon the

traditional interactive methods. First, however, because they formed the backbone of most MOLP solution theory, the prior and posteriori approaches merit some consideration.

3.2 Prior articulation methods

3.2.1 Preemptive Goal Programming

Preemptive linear goal programming (Lee, 1973) may be the most recognized MOLP solution technique regardless of classification. Although Lee is given most of the credit for the practical development and applications of goal programming, its foundations were laid by Charnes and Cooper (1961) and Ijiri (1965). In short, goal programming attempts to minimize a set of deviations from DM specified multiple goals for mathematically defined objectives. The goals are considered simultaneously but are weighted according to their relative importance. Essentially, these preemptive priorities invoke a ranking system for the deviational variables such that only the highest priority goals are solved first. Subsequently, the second highest priority goals are met as closely as possible (deviations minimized) without disturbing the higher priority goals. The process continues through the lowest priority objectives. Since its development, several variations of goal programming and goal programming solution techniques have appeared (see Ravindran (1985) and Arthur and Ravindran (1978)). However, the preemptive formulation proposed by Lee remains the most recognized in the field.

3.2.2 Maximin Programming and Surveys

Essentially, preemptive linear goal programming is the only

prior articulation method used in practice for solving MOLPs. Another lesser known technique is that of Maximin Programming. Although this technique will not be expounded here, further information can be found in Dyson (1980). For the interested reader, surveys of prior articulation techniques for solving general MCDM problems, linear or not, can be found in Dyer and Sarin (1979), Farquhar (1977), Keeney and Raiffa (1976), and Kornbluth (1973).

3.2.3 **Critique of prior articulation techniques**

Despite their early contributions to MCDM, the prior articulation methods are not as practical as once thought. Obviously, by requiring DM input of goals prior to any preliminary computations, a DM is often left in a position of setting "ignorant" aspirations. With no information regarding the range of potentially achievable solutions, it is highly unlikely DM aspirations would approach an efficient alternative. This often occurs in linear goal programming as the deviational variables either assume all zero (a dominated alternative) or all nonzero (i.e. no goals are achievable) values. While it is true that several iterations of prior methods could eventually lead to an efficient solution (note this would actually entail an interactive method), there is still no semblance of convergence to a DM specified best compromise solution.

3.3 **Posteriori articulation methods**

The foundation for much MOLP research was built upon techniques requiring a posteriori preference articulation.

Algorithms which employ this type of strategy can generally be divided into two categories: 1) those concentrating on finding all efficient extreme points, and 2) those which concentrate on finding the entire efficient set. It has been established (Yu, 1985) that the set of efficient points of a given decision space X lies on the boundary of X assuming linear independence of the objective functions. Since the faces (facets, edges, points) can be characterized by the extreme points and/or extreme rays of X, it can be shown that if an interior point of a face is efficient, then the whole face is efficient (Yu, 1985). Thus, the entire efficient set is connected and the entire efficient domain can be explored point by point without ever leaving the set. As a result, MOLPs tend to lend themselves to simplex based techniques in order to identify efficient extreme points. Given these extreme points, several techniques have been suggested for connecting these points and thus constructing the efficient frontier.

3.3.1 Category 1: **Finding all efficient extreme points**

In general, algorithms in category (1) consist of three phases (Steuer, 1976a). Phases I and II deal with finding an initial extreme point and subsequent initial efficient extreme point. Phase III then identifies the remaining efficient extreme points.

3.3.1.1 **The Multiobjective Simplex Methods**

The best known of these "efficient extreme point generators" are the multiobjective simplex methods. Variations of these algorithms have been proposed by Philip (1972), Evans and Steuer (1973), Yu and Zeleny (1975), and Zeleny (1982). Multiobjective

simplex techniques require an initial efficient extreme point for initialization. Several techniques for finding an initial efficient extreme point have been developed (Ecker and Kouada (1975) , Evans and Steuer, (1973) , and Zeleny, (1974a)). Once this point is identified, the simplex algorithm is used to pivot through adjacent extreme points using special conditions (tests) to identify those which are efficient.

3.3.1.2 **Parametric Decomposition (a "weighting" technique)**

Because of the computational burden of multiobjective simplex tableaus, a variation of the multiobjective simplex methods known as parametric decomposition was developed in 1973 by Zeleny (see Zeleny, 1982). Instead of optimizing the several objectives separately, the objectives are combined into a normalized weighted objective function. In the case of MOLP, maximizing this objective function for all possible weighting coefficients reveals the entire set of efficient extreme points. Thus, the parametric weight space is "decomposed" into subsets associated with nondominated solutions.

3.3.1.3 **Other weighting techniques**

An approach proposed by Gal (1976), Ecker and Kouada (1978), and Ecker et. al. (1980) makes explicit use of weighting characterizations to identify noninferior extreme points. Given two adjacent extreme points to MOLP, these researchers have shown that the alternatives are adjacent <u>efficient</u> extreme points iff 1) they are adjacent basic feasible solutions, and 2) they are both optimal solutions to the same weighted objective function for some

set of positive weights.

3.3.2 Category 2: Finding the entire efficient set

The more recent work in the posteriori class has concentrated on algorithms for generating the entire efficient set. The reason for this is clear. In most instances, the best compromise solution will be an efficient point which is not an extreme point. Thus, these methods concentrate on finding the maximal efficient faces of the constraint set. Unfortunately, these methods are not as computationally straightforward as extreme point identification. Sage (1977) uses information about pairwise adjacency of extreme points to construct undirected graphs of binary intersection matrices. Applications of this approach can be found in Zeleny (1974a), Yu and Zeleny (1975), and Isermann (1977, 1979). As reported by Chankong and Haimes (1983), it is also possible to use weighting methods to identify noninferior faces. For the interested reader, other work in this area has been accomplished by Ecker et. al. (1980) and Yu and Zeleny (1975).

3.3.3 Critique of posteriori techniques

Although interesting in their theoretical base, the a posteriori techniques suffer from two major shortcomings. First, computation of all efficient faces (or even extreme points) can become quite impractical. Theoretically, an exponential number of extreme points can arise. Furthermore, given that all efficient faces or extreme points can be identified, presentation of these solutions to a DM in a manner such that he or she can choose among them can prove to be an even more difficult undertaking.

3.4 Interactive techniques for MOLP

Many researchers believe that interactive methods may be the answer to weaknesses characteristic of the prior and posteriori methods of sections 3.2.3 and 3.3.3. Traditional interactive algorithms for solving the MOLP are discussed in the sections which follow along with brief critiques of each method.

3.4.1 The Interval Criterion Weights Method

The Interval Criterion Weights method (ICW) (Steuer (1976b)) involves DM comparison of 2p+1 efficient extreme point solutions at each iteration where p is the number of objectives. The DM must choose a most preferred of these solutions. Based upon DM election, the weight space (criterion cone) is reduced, and a new set of 2p+1 solutions is generated for further comparison. Although commendable for its avoidance of requiring a DM to place specific weights on objectives, ICW can prove burdensome for a DM with respect to the number of comparisons required at a given iteration (i.e. if p is large). A second deficiency results from the fact that only efficient extreme points are considered. Thus, efficient alternatives lying on higher dimensional faces are never presented to the DM for consideration.

3.4.2 Displaced Ideal Method

In the Method of the Displaced Ideal (Zeleny, 1974b), a set of efficient solutions is reduced at each iteration until it is small enough for the DM to feel comfortable in choosing one as a best compromise solution. This is accomplished by removing each solution which results in an outcome which is at a certain distance

or further from the "displaced ideal". As with the ICW method, the DM may be confronted with multiple comparisons of similar alternatives. Because the method assumes DM ability to choose one alternative from the set, the algorithm stalls if a DM is unable to consistently perform the comparison task.

3.4.3 **Interactive Paired Comparison Simplex Method**

Another more recently developed technique implementing DM comparison of efficient extreme points is the Interactive Paired Comparison Simplex Method (Malakooti and Ravindran, 1985). Similar to other methods, an initial efficient extreme point is generated. The algorithm then pivots to another efficient extreme point and requests the DM comparison of this alternative with the current point. If the DM prefers the adjacent point, it becomes a "center solution", the initial point is eliminated, and the algorithm continues. Otherwise, the adjacent point is eliminated. The algorithm terminates at a best compromise solution when there are no efficient adjacent extreme points preferred to the current point. The main contribution of this method was the development of a utility efficiency concept, used to eliminate some adjacent points without requiring DM comparison. Unfortunately, as is the case with many simplex based algorithms, only extreme points are considered. Although this technique also assumes DM ability to make comparisons (like ICW and Displace Ideal), the feature of requiring only a paired (two at a time) comparison of points makes the method practical and less burdensome from the DM standpoint. In addition, a method for handling any inconsistent DM responses

was also incorporated.

3.4.4 **The Method of Zionts and Wallenius**

A now famous interactive technique for MOLP is the method of Zionts and Wallenius (1976). The algorithm employs a pairwise comparison technique for systematically evaluating efficient extreme point solutions. After an efficient extreme point is attained (usually via a method alluded to in section 3.3.1.1), a DM is asked to compare it to an adjacent extreme point via tradeoff preference inquiries. Based upon DM responses, a linear approximation of the DM's utility function is constructed. At each iteration, the linear approximation is improved until a satisfactory solution is achieved. Again, disadvantages appear in that only extreme points are considered. In addition, the method of interaction with the DM (tradeoff inquiries) can prove difficult resulting in DM confusion and inconsistency. Furthermore, convergence is only guaranteed if a DM's utility function is indeed linear.

3.5 **Aspiration based techniques**

As previously alluded, a special class of interactive methods are of special interest to this research. These are the aspiration level based techniques. Two of the more noteworthy traditional methods are discussed below.

3.5.1 **STEM/GPSTEM**

One of the first techniques to address MOLPs via progressive articulation of preferences and DM aspiration levels was the Step Method (STEM) of Benayoun et. al. (1971). STEM employs a single

objective model which minimizes the maximum weighted distance of all problem objectives from the "ideal" solution. At each subsequent iteration, the DM "adjusts" the feasible region by adjusting the aspiration levels for the objectives.

A variation of STEM, known as GPSTEM, was developed by Fichefet (1976). GPSTEM combines linear goal programming and STEM in that at each iteration, a goal program is employed as the single objective model.

Obviously, these simplex-based techniques are only capable of divulging extreme point alternatives.

3.5.2 **The Method of Wierzbicki (Tchebycheff methods)**

One of the most commonly used techniques for generating efficient solutions for comparison in modern algorithms is the scalarizing approach of Wierzbicki (1979). In and of itself, this approach requires prior preference articulation. However, because of its ability to achieve any alternative on the efficient frontier (extreme point or not), it is often employed to project aspiration levels of a DM onto the efficient set. These projections are then used as the basis for comparisons in interactive techniques. In fact, the method of Wierzbicki is employed as the "efficient point generator" for both the AIM and CASE methods discussed in section 4.

The Wierzbicki technique makes use of a penalty scalarizing function to compute aspiration level projections regardless of their feasibility. Advantages of this approach include the fact that it is not simplex based and thus is not confined to extreme

points. Furthermore, no assumptions are made about the so called utility function of the DM. On the negative side, the method generally works well only when aspiration levels are "near" the efficient frontier. Assuming a DM has no concept of the efficient set, the Wierzbicki method may produce an undesirable "nearest nondominated solution" to the aspirations. To compensate, interactive methods using the Wierzbicki technique may require a large number of iterations for satisfactory convergence.

3.6 **Conclusion**

While it is true that CAIN will implement variations of concepts developed by traditional techniques, initial motivation for CAIN development stemmed from the newly developed aspiration based methods. In fact, some of the more practical features of CAIN are based upon concepts employed in AIM. Section 4 therefore presents a detailed review of modern aspiration based algorithms. In addition to a detailed discussion of the discrete AIM technique (section 4.1), subsequent sections are devoted to previously developed Tchebycheff interactive strategies for MOLP.

4.0 MODERN INTERACTIVE ALGORITHMS

4.1 AIM - An Aspiration level Interactive Method

4.1.1 Introduction

AIM (Lofti, Stewart, and Zionts, 1990) is described by its developers as a simple eclectic approach for solving the discrete alternative MCDM problem. As the term discrete implies, the method considers the specific problem of choosing one alternative from a finite set of defined alternatives. A DM has a set of objectives which describe each of these alternatives and each alternative has a specific measure of achievement for each of these objectives. DM levels of aspiration are used to explore the efficient frontier with AIM providing feedback as to the feasibility of such aspirations. Specifically, the fraction of all solutions that satisfy current aspiration levels is provided considering all objectives collectively as well one at a time. The user-friendliness of AIM is further enhanced by suggestions of "nearby solutions" and by information concerning attainment possibilities of various alternative objective levels.

4.1.2 Philosophy and notation

AIM assumes a matrix (n x p) of alternatives and objectives where n represents the total number of alternatives and p represents the number of objectives. Entries in the matrix consist of the performance of alternative i with respect to objective j. A generalized form of the matrix is shown in fig. 4.1.

fig. 4.1

$$\begin{bmatrix} P_{11} & P_{12} & \cdots\cdots & P_{1p} \\ P_{21} & P_{22} & \cdots\cdots & P_{2p} \\ \cdot & \cdot & \cdots\cdots & \cdot \\ \cdot & \cdot & \cdots\cdots & \cdot \\ P_{n1} & P_{n2} & \cdots\cdots & P_{np} \end{bmatrix}$$

where P_{ij} represents performance of alternative i for objective j

AIM permits both cardinal and ordinal objectives (i.e. objective and subjective scales) and three types of objective functions:

1) those to be maximized

2) those to be minimized

3) those which have a target level

A noteworthy feature of AIM is incorporation of the concept of satisficing levels. With each type of objective function, a DM is permitted to set satisficing thresholds. As defined for AIM, a satisficing threshold is a level where the DM is "effectively indifferent to values above (for maximization), below (for minimization), or within a certain range (for target)". These thresholds permit the DM the flexibility not to overachieve for any objective for which a satisficing threshold is defined.

Given an MCDM problem definition in terms of objectives (of the three possible types above, with or without satisficing thresholds), a finite set of alternatives, and performances of each alternative (on cardinal or ordinal scales) for each objective, the algorithm can begin. For each objective, an absolute or "must" level is assumed which must be met as well as an aspiration or

"want" level and, if desired, a satisficing or "ignore" threshold. Ordinal objectives must be given ordinal values of which only the order of the values is essential. Furthermore, it assumed that objectives with a target value can be represented as two objectives; one maximizing to achieve the lower target value, and one minimizing to achieve the upper target value. Clearly, the lower end of the target range cannot exceed the upper end of the range. For ease of describing the algorithm, the authors define the following notation:

T_i: satisficing threshold for objective i.

z_i^k: value of alternative k in terms of objective i.

I_i: ideal value for objective i according to DM aspirations: $\min[T_i; \max_k\{z_i^k\}]$ in the maximization case, $\max[T_i; \min_k\{z_i^k\}]$ in the minimization case.

N_i: nadir, or worst possible value, for objective i: $\min_k\{z_i^k\}$ for maximization; $\max_k\{z_i^k\}$ for minimization.

A_i: aspiration level for objective i which should not exceed I_i.

4.1.3 **Solution Methodology**

AIM solution methodology begins by ordering the z_i^k values from the least to the most preferred. The DM is then presented with the following information:

1) A_j, the current aspiration level for objective i along with the percentage of all alternatives that are at least as desirable with respect to this objective. Initially, aspiration levels are set to the median

for each objective given the ordering of the z_i^k values.

2) aspiration levels <u>one level different</u> from those presented in 1), both one better and one worse. Again the percentage of all alternatives at least as desirable as these values are shown.

3) ideal and nadir values for each objective.

4) the proportion of alternatives that simultaneously satisfy all aspiration levels given for both 1) and 2).

5) a "nearest" efficient solution to DM aspiration levels as defined by a penalty scalarizing function proposed Wierzbicki. Reference to the method of Wierzbicki is given in the literature review (section 3.5.2).

The Wierzbicki scalarizing function gives a weight on criterion i of:

$$\frac{A_i - N_i}{I_i - N_i} \quad \forall i=1,2,\ldots,p$$

An examination of these weights reveal the increasing importance attached to objective i as the aspiration level for i is moved closer to the ideal value. More specifically, the solution which minimizes the following penalty scalarizing function is chosen as the "nearest" efficient solution:

$$Max_i \{ d_i \} + e \sum_{i=1}^{p} d_i$$

where

$$d_i = \frac{(A_i - N_i)(A_i - z_i^k)}{(I_i - N_i)^2}$$

At this stage, several options are available to the user:

1) Update current aspiration levels. Only realizable values for each objective are allowed. Note that as these levels are changed, so too is the nearest efficient solution.
2) Scan all solutions satisfying current aspiration levels.
3) Rank the current alternatives according to the weights specified for the scalarizing function given above.
4) Scan neighboring solutions. Here the method uses a simplified version of the ELECTRE method (Roy, 1968): Alternative j outranks alternative k iff:
   i) the fraction of objectives for which j is at least as good as k is at least 50%.
   ii) the following condition is satisfied:

$$Max_i \left[\frac{z_i^k - z_i^j}{I_i - N_i} \right] \le c$$

where c is determined by the number of neighbor solutions desired

5) Lastly, the DM may explore the distribution of each of the problem objectives.

4.1.4 **Experimentation with AIM**

The AIM algorithm has been tested and compared against another discrete alternative MCDM method, the Analytic Hierarchy Process (AHP). According to the report, AIM outperformed the AHP "for numerous measures". For the experiment, 49 second-year graduate students were asked to solve two decision problems using both a computer implementation of AIM and Expert Choice, a computer implementation of AHP. Using AIM, the authors report that 98% of the participants chose efficient solutions versus 91% for Expert Choice and 79% for a manual decision process. The differences proved to be statistically significant. Other measures included agreement between algorithm choice and a students' final decision, and attitudes toward the methods. Again, the authors report AIM superiority to the AHP. A detailed description of the experimental phase of AIM can be found in Lofti, Stewart, and Zionts (1990).

4.1.5 **Critique of AIM**

While AIM has implemented several useful features which could be useful in a continuous MOLP solution approach, it is in many ways confined to the discrete problem domain. Many of the

computations performed and information presented by AIM are not applicable to continuous MOLPs. For instance, information concerning "percentage of desirable alternatives" and "ranking of alternatives" is not practical when the alternative set is infinite.

On a more theoretical note, AIM also suffers from a common interactive approach deficiency. Specifically, there is no assurance of convergence to an efficient solution. Wierzbicki (1979) stated that it is not so important for a DM to select nondominated solutions, but simply that he or she be aware of their existence. Unfortunately, the common everyday decision maker may not fully comprehend the concept of nondominance. For discrete problems, simply alerting a DM to the existence of efficient alternatives may be sufficient. However, in the realm of the continuous domain, the number of potential alternatives can be overwhelming. Therefore, it is the opinion of this author that for continuous MOLPs, only efficient alternatives should be presented to a DM.

On a different front, some thought must be given to the method of arriving at a nearest nondominated solution. The AIM developers make reference to the fact that DM aspiration levels should be "near the efficient frontier" in order for the Wierzbicki method to have "its greatest effect". Unfortunately, it cannot be assumed that a potential DM is aware of the efficiency thresholds.

On a more positive note, AIM has implemented some useful concepts for reducing the burden placed on a DM. The concept of

requesting aspiration levels rather than often difficult comparisons between alternatives (which often leads to DM inconsistency) or tradeoff questions is notable. In addition, the implementation of satisficing levels gives a DM the opportunity to realize further gains in other objectives once a given objective is known to have reached the DM's utility saturation point.

In reality, although AIM employs features useful for any MCDM solution technique, its discrete domain renders AIM and CAIN incomparable. In sections 4.2 - 4.4, however, three modern algorithms not confined by extreme points or discrete alternatives are reviewed for eventual subjective comparison with CAIN. Section 4.5 then concludes the literature reviews with some observations and a brief introduction to interior point approaches.

4.2 CASE - A Convergent Aspiration based SEarch method

The original intent of CASE (Yoon, Lofti, and Zionts, 1991) research was to develop a continuous linear domain extension of the discrete AIM methodology described in section 4.1. Given an MOLP formulation in the decision space (see section 2.1) with p objectives, the CASE algorithm generates a cluster of p+1 alternatives at each iteration based upon DM levels of aspiration. The method assumes a pseudoconcave utility structure for a DM throughout the interactive process. Assuming a DM can choose a preferred solution from the cluster of p+1, CASE eventually reduces ("derives cuts in") the feasible objective space based upon DM responses to the comparisons. Through successive feasible set reductions, CASE theoretically converges to a DM aspired best

compromise solution.

CASE incorporates the concepts of ideal and nadir solutions to guide DM aspiration levels. Given a defined feasible region, ideal and nadir solutions are generated for every objective. A DM must then specify aspiration levels within this range of values. Based upon the aspirations, CASE generates a nearest nondominated solution on the efficient frontier using the Wierzbicki scalarizing technique. This solution becomes a "center" solution. Concurrently, the system generates p "neighbor" alternatives on the efficient frontier by using the concept of improving direction vectors. These solutions are the ultimate result of improving one objective at a time from the center alternative A DM must then select a preferred alternative from the cluster of p+1. If a DM selects one of the neighbor solutions, this alternative becomes the center solution and another set of p neighbors are generated for comparison. This process continues until a DM selects the center solution as the most preferred. At this time, CASE generates a cut in the feasible objective space and thus reduces the feasible region. Theory behind the cuts rests upon utility concepts. When a DM chooses the center solution as most preferred, the information is interpreted as $U(z^0) \geq U(z^0 + d^j) \; \forall j=1,2,\ldots,p$ where z^0 is the center solution and d^j, j=1,2,...p represent improving directions for each objective. This information in combination with the assumption of DM pseudoconcave utility permit the feasible region reductions(for theoretical details, see Yoon, Lofti, and Zionts, (1991)). CASE is terminated when the reduced objective space is

considered "small enough", if a predetermined number of iterations has been reached, or if a DM is satisfied with the chosen solution.

4.3 **Karwan-Dell (KD) Method**

The KD (Dell and Karwan, 1989) method uses predefined weights in a specialized form of Tchebycheff LP formulation to generate nondominated alternatives. The method is based upon pairwise comparison of alternatives and uses constraints generated on the weight space as the method of feasible region reduction. This method is similar to the Z-W algorithm described in section 3.4.4 except that the specialized LP formulation allows generation of nonextreme point solutions.

Initially, the system determines the ideal solution for each objective. Two solutions are then generated, a "challenger" and an "incumbent". The system then asks a DM to compare and choose one solution from these two alternatives. Based upon the response, the system generates constraints on the weight space. Unlike Z-W however, these constraints partition the feasible space into multiple disjoint convex regions in the weight space. For each of these regions in the weight space, KD obtains a middle-most set of weights and from these weights, a nondominated alternative is generated. The method then groups these alternatives and chooses the new challenger as an alternative that is significantly different from the incumbent. The algorithm terminates if no significantly different alternatives exist or the system has exceeded a specified number of questions with the incumbent as the most desired alternative.

4.4 **Steuer-Choo (SC) Method**

The SC (Steuer and Choo, 1983) algorithm is another method centered around a weighted Tchebycheff procedure. SC samples the efficient set by computing the efficient vector closest to an ideal according to the weighted function. Using a filtering technique, representatives of smaller and smaller subsets of the efficient frontier are presented to a DM at each iteration. The procedure continues to generate sets of weighting vectors within a contracted cone and subsequently generates sets of efficient solutions until a prespecified number of iterations is reached. Like KD, the SC technique is essentially an extension of an earlier simplex based method (see ICW, section 3.4.3) The extension allows for consideration of nonextreme point solutions.

4.5 **Observations and Introduction to Interior point methods**

Clearly, while traditional methods are confined by simplex ideology, the new approaches appear centered upon Tchebycheff or scalarizing techniques for generation of efficient solutions. While this signifies an improvement in the domain of achievable solutions, DM interaction techniques appear to have remained intact over the years. Interactive scalarizing techniques ultimately require DM comparisons of alternatives and their convergence is contingent upon DM ability to consistently perform this task.

The main purpose for the review of sections 3 and 4 was not so much to search for techniques to implement as part of a new aspiration based algorithm, but rather to explore and shed light on the current state of MOLP solution methodologies. Until recently,

virtually all techniques employed in the solution of MOLPs made use of some form of the simplex method by first converting the MOLP to one or a series of single objective linear programs and using a system of weights to produce efficient points. This direction was undoubtedly and rightfully spawned by the practicality and computational efficiency of simplex developments of the 1960's and 1970's. However, because of its confinement to extreme points and the computational complexity of identifying efficient faces, modern research has begun to explore other optimization options. Today, a new strategy of optimization is generating enormous interest - the interior point approach. Although Klee and Minty (1972) demonstrated that simplex based approaches could theoretically require an exponential number of pivots (and thus require exponential time), the computational complexity of linear programming stagnated the optimization field until 1979. In 1979, Khachian showed that it was possible to solve LP's in polynomial time by using an ellipsoid algorithm. Essentially, the ellipsoid algorithm approximates the linear constraint set with mathematically well behaved continuous curves. A famous interior point approach by Karmarkar (1984) and others (Marsten, McShane, etc.) have since demonstrated the computational effectiveness of interior point approaches to the single objective LP.

Application of interior point techniques to MOLPs remained relatively unexplored until 1989 when the well established method of centers (Huard, 1967) was applied to vector optimization (Morin and Trafalis, 1989). Given an interior point of a polytope and an

"ordering cone" of DM preferences, it was shown that a progressive method of centers converges to an efficient point (Trafalis, 1989). As the algorithm of centers is key to CAIN, it shall be discussed in greater detail during CAIN development.

In the development of CAIN, every attempt was made to incorporate the positive features of previously developed methods. Concepts such as ideal and nadir solutions, absolute requirements, satisficing levels, and aspiration levels all form an integral part of CAIN. On the other hand, while there are many similarities, it is hoped that there have also been significant improvements. New techniques have been incorporated in the areas of DM interaction, efficient solution generation, and convergence. In section 5 which follows, a detailed development of the CAIN algorithm is presented.

5.0 A CONVERGENT ASPIRATION BASED INTERIOR METHOD (CAIN) FOR MOLP

5.1 Introduction

Like AIM, development of CAIN was motivated by the need for a simple, practical approach for guiding a DM to an efficient best compromise solution considering conflicting objectives. Without proceeding further, the terms "simple" and "practical" are often overused without explanation to describe many newly developed algorithms. In describing CAIN, "simple" makes reference to computational complexity and theoretical convergence. With the exception of the mathematical development of the Algorithm of Centers, CAIN is computationally straightforward. Only solutions of single objective linear programs are required to initiate CAIN convergence to efficiency. From a "practicality" standpoint, a DM must only furnish aspiration levels, and is kept fully aware at all times of the achievable range of values for each objective. Through an innovative interaction technique, CAIN forces DM aspiration levels to remain feasible, yet only presents efficient solutions to a DM for final consideration. It also incorporates the ideas of satisficing levels and absolute minimum requirements employed in AIM.

CAIN is subdivided into three phases: Initialization, Interaction, and Convergence. After an introduction to the CAIN problem domain and a new DM interaction technique (sections 5.2 and 5.3), these three phases are explained in detail in sections 5.4 - 5.6. Section 5.7 discusses the theoretical convergence of CAIN. Finally, sections 5.8 and 5.9 present a formal development of the

algorithm complete with flowcharts for each phase.

5.2 Interaction Domain

CAIN is designed for DM interaction in the objective space. Given a problem definition in terms of decision variables (see equations (1) and (2) in section 2.1), it must be redefined (mapped) in terms of the objective space (see section 2.3). After completion of this transformation, DM interaction is necessary for algorithm initialization. The following section describes a new DM interaction strategy developed for the CAIN algorithm.

5.3 DM interaction using an Aspiration Level Range Method (ALaRM)

A unique feature of CAIN is its method of interaction with a DM. The technique, known as ALaRM (Aspiration Level Range Method), is based upon concepts of ideal and nadir values (see section 2.5) and works as follows.

Given a feasible region in the objective space (U), ideal and nadir values for a given objective f_i can be computed and presented to a DM. Using these values as upper and lower bounds, a DM is requested to furnish a desired level for f_i. Normally, this specified value will reduce the range of achievable values for other objectives from their respective ideal and nadir values initially achievable subject to region U alone. Therefore, before requesting DM desires for another objective f_j, new ideal and nadir values are computed for f_j subject to region U <u>and aspirations already furnished for other objectives during the current iteration</u>. This process continues until all objectives are considered.

For instance, consider the feasible objective space given by the example of section 2.4 and shown graphically in fig 5.1.

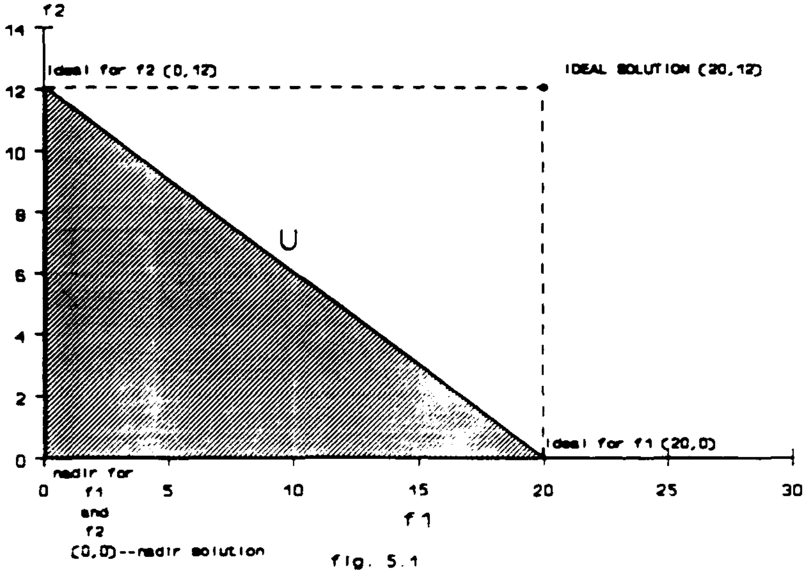


fig. 5.1

Assume a DM desires $f_1 = 10$. Since the maximization convention is assumed, any alternative with $f_1 \geq 10$ is deemed acceptable (assuming no satisficing level for f_1 exists). Given this aspiration level constraint for f_1, the reduced feasible space is sho.n in fig 5.2.

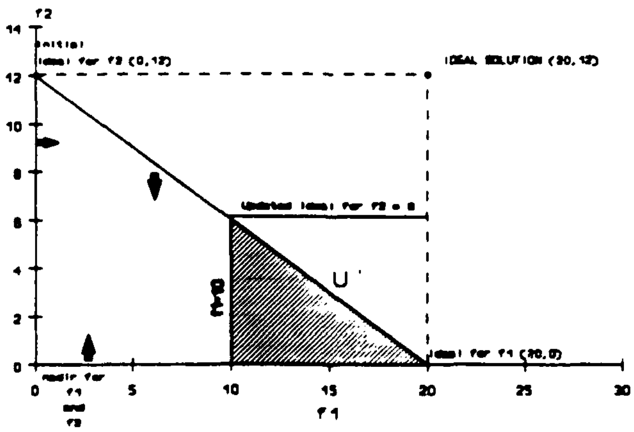


fig. 5.2

Alarm recomputes ideal and nadir ranges for f_2 based upon this updated feasible space (U′) <u>before</u> requesting further aspirations from a DM. Clearly, the updated ideal value for $f_2 = 6$, down from its initial potential value of 12. Alarm would then request an aspiration level for f_2 within the range [0,6]. Note that any point in this range along the line $f_1 = 10$ remains feasible, thus rendering DM aspirations feasible.

The major advantages of Alarm are twofold. First, by continually updating ideal and nadir values, DM aspirations are forced to remain feasible. Because the Algorithm of Centers requires an interior point of the defined feasible region for initialization, this feature will prove key to CAIN convergence. Secondly, a DM is constantly kept aware of the achievable range of values for all objectives at all times. This differs from interaction techniques of AIM and CASE where DM aspiration levels are requested simultaneously. Only after the Wierzbicki function generates a nondominated solution is a DM made aware of the feasibility of his or her aspirations. By proceeding to identify a nearest nondominated for potentially unachievable aspirations, a DM has no input as to which objectives are compromised to arrive at the efficient frontier. As will be seen later, CAIN not only keeps a DM updated with regard to achievable objective levels, but also permits a DM the flexibility to choose the order in which aspirations are to be considered.

5.4 <u>**Initialization Phase**</u>

CAIN initialization requires DM input of minimum requirements

and any known satisficing levels for each objective. This interaction follows the Alarm interaction technique of the previous section. Once these levels are properly defined, initialization is complete. Minimum requirements and satisficing levels remain rigid throughout the algorithm and are used solely for initialization purposes. Later, it shall become obvious that these constraints almost immediately become redundant during the interaction phase.

5.5 **Interaction Phase**

Two types of DM specifications are permitted and/or required during the interaction phase:

1) First, a DM is given the option of specifying priority rankings for the objectives in terms of their relative importance. Three possible methods of specification are possible:

i) a DM can rank each objective individually as a unique priority (given p objectives, there will be p separate rankings).

ii) a DM can establish groups of priorities (given p objectives, there will be less than p ranking groups).

iii) a DM can choose not to rank the alternatives (in which case the objective are given the same priority and are treated in no particular order).

2) Aspiration levels for the objectives are the only required input from a DM. These levels are requested in the order of the defined priorities. Again, the Alarm method is employed as the interaction tool. Ideal and nadir values are computed based upon

the original feasible objective space, the defined minimum requirements, any satisficing levels, and any aspiration levels already defined for other objectives during the current iteration. Recall that Alarm forces a DM to select aspiration levels between the respective ideal and nadir values for each objective. As a result, the levels remain feasible.

A complete iteration consists of successful entry of aspirations for each objective. In the interest of flexibility, CAIN allows for DM reconsideration of aspirations within iterations. Specifically, if a DM deems the range of ideal and nadir values for a particular objective unacceptable, he or she is permitted lower aspirations for the higher priority objectives previously defined during the current iteration. Obviously, the hope here is to broaden the range of achievable values for the objective currently under consideration. Furthermore, at the start of new iterations, a DM is permitted to redefine objective priority rankings in order to place emphasis on objectives which may not be progressing satisfactorily.

5.6 **Convergence phase**

Upon successful completion of an iteration, the Algorithm of Centers is invoked and converges to a "best compromise solution" based upon DM defined levels of aspiration. This solution is presented to the DM. He or she may either accept this solution and terminate the algorithm or begin another iteration of defining tighter aspiration levels.

5.7 **Observations**

A noteworthy observation of CAIN is its amicable nature from a DM standpoint. A DM need only furnish aspiration levels for objectives to initiate convergence to a best compromise solution. Ranking of objectives is permitted, if desired, but not required. In this light, CAIN may be described as a form of aspiration based interior point goal programming. Note that no assumptions are made concerning a DM's ability to make comparisons between alternatives or respond to difficult tradeoff inquiries. Such burdensome requirements were purposely avoided. Furthermore, no implicit assumptions concerning a DM's utility function are needed. As long as a DM can choose a desired value from a given range of values for the each objective, the algorithm will converge. Toward the development of a convergent algorithm, some necessary notation and definitions are presented in the next section. Section 5.8 follows with a discussion of CAIN convergence. Section 5.9 then presents a formal development of the CAIN algorithm.

5.7 **Definitions**

Before formally introducing CAIN, some additional notation is necessary.

Definition 5.7.1: *Let R represent a set of absolute level constraints imposed by a DM; that is, if r_i represents the absolute minimum requirement for objective f_i, $i=1,2,\ldots,p$, then*

$$R=\{f \mid f_i \geq r_i \quad \forall i=1,2,\ldots,p\}$$

Definition 5.7.2: *Let S represent a set of satisficing level*

constraints imposed by a DM; that is, if s_i represents the satisficing level for objective f_i, $i=1,2,\ldots,p$, then

$$S = \{ f \mid f_i \le s_i \quad \textit{for some } i=1,2,\ldots,p \}$$

In simple terms, a satisficing level s_i can be thought of as a threshold beyond which a DM gains no further satisfaction from increasing objective f_i. By not allowing objective f_i to exceed s_i, higher levels of other objectives f_j, $i \neq j$ can be realized. Unlike set R, set S need not be defined for every objective f_i (thus the use of "some" in the definition). An aggressive DM would most certainly have minimum requirements for every objective. However, this may not be true of satisficing levels. In many cases, it is likely a DM may wish to maximize all objectives as much as possible.

<u>Definition 5.7.3</u>: *Let $A^{(k)}$ represent a set of aspiration level constraints imposed by a DM at the kth iteration; that is, if a set $W^{(k)}$ represents DM aspiration levels y_i for objectives f_i, $i=1,2,\ldots,p$ at iteration k, then*

$$\begin{aligned} & W^{(k)} = \{ f \mid f_i = y_i \quad \forall i=1,2,\ldots,p \} \\ \textit{and} \quad & A^{(k)} = \{ f \mid f_i \ge y_i \quad \forall i=1,2,\ldots,p \} \end{aligned}$$

where $W_i^{(k)}$ represents the aspiration level and $A_i^{(k)}$ represents the aspiration level constraint for objective f_i at iteration k.

Clearly, if the DM desires a level y_i for objective f_i, then

under the maximization convention, any alternative where $f_i \leq y_i \ \forall i=1,2,\ldots,p$ should be eliminated from further consideration during iteration k.

There is no discernable difference in the way sets R, S, and set $A^{(k)}$ are imposed within the algorithm. Sets R and S are used exclusively to initialize CAIN and remain fixed throughout the interactive and convergence phases. Set $A^{(k)}$ on the other hand is flexible as a DM articulates tighter aspiration levels for successive iterations. As shall be shown later, set S also serves to aid in the definition of the "ideal" values for any objectives for which set S is defined.

Observe that for a typical iteration, there may exist as many as m+n+3p objective space constraints defined by sets R, S, and $A^{(k)}$. The possibility of m+n constraints result from the m real and n nonnegativity constraints mapped from X to F (although often, some of these mapped constraints prove redundant). The 3p constraints can result from sets R, $A^{(k)}$, and any constraints defined for set S. Clearly, constraints from sets R and S will also prove redundant as aspiration levels tighten the feasible space. They therefore can and should be eliminated whenever possible in the interest of reducing problem size.

Although the notion of "ideal" and "nadir" solutions for MOLP are not unique to the CAIN algorithm (see section 2.5), specialized definitions of these concepts must be made due to incorporation of satisficing levels. Specifically,

<u>Definition 5.7.4</u>: An <u>ideal</u> value for a particular

objective f_i during iteration k, $I_i^{(k)}$, is the maximum value of f_i attainable subject to sets F, R, S, and $A^{(k)}$. In formal notation:

$$I_i^{(k)} = \begin{matrix} \mathit{Max}\ f_i \\ s.t.\ f \in (F \cap R \cap S \cap A^{(k)}) \end{matrix} \quad \forall i=1,2,\ldots,p$$

<u>Definition 5.7.5</u>: A <u>nadir</u> value for a particular objective f_i during iteration k, $N_i^{(k)}$, is the minimum value of f_i attainable subject to sets F, R, and $A^{(k)}$. In formal notation:

$$N_i^{(k)} = \begin{matrix} \mathit{Min}\ f_i \\ s.t.\ f \in (F \cap R \cap A^{(k)}) \end{matrix} \quad \forall i=1,2,\ldots,p$$

Throughout CAIN, a DM will be presented the option of defining priority rankings for the objective functions with regard to their relative importance. More specifically, a DM shall be permitted the following options:

1) Rank each objective function individually (q=1,2,...,p)
2) Rank the objectives in groups (q=1,2,...<p)
3) Do not rank the objectives (q=1: all objectives have the same priority).

Toward this end, let:

$$P_q \doteq \begin{matrix} \textit{a ranking system} \\ \textit{where } 1 \le q \le p. \end{matrix}$$

Furthermore, let P_1 represent the set of objective functions ranked as most important, P_2 represent the "priority 2" objectives...and P_q represent the least important objectives.

Definition 5.7.6: *Define a set of indices J_q as follows:*

$$J_q = \left\{ \begin{matrix} i \mid f_i \in P_q \\ \forall i=1,2,\ldots,p \\ q=1,2,\ldots,v \end{matrix} \right\}$$

where $v \leq p$ represents the number of different ranking groups specified by a DM for the objective functions.

Definition 5.7.7: *A complete iteration k occurs when a DM has successfully established levels for set $A^{(k)}$ for all existing objectives.*

Upon successful completion of an iteration, CAIN must converge to an efficient "best compromise solution". Toward this goal, define an "ordering cone" at iteration k as follows:

Definition 5.7.8: *An ordering cone of DM preferences based upon DM aspiration levels at iteration k is defined as:*

$$\Lambda^{(k)} = \left\{ f \mid f \in \bigcap_{i=1}^{p} A_i^{(k)} \right\}$$

Given an interior point $W^{(k)}$ of F and the ordering cone at iteration k, it can be shown that the following sequence of steps (known as the Algorithm of Centers) will converge to an efficient solution $f^{(k)*}$ (for proof, see Trafalis, 1989). Furthermore, since $W^{(k)}$ is specified by a DM and defines the ordering cone, $f^{(k)*}$ can be considered an efficient best compromise solution at iteration k.

0. Set z=0

1. Let $W^{(k)z}$ be an interior point of F. Consider the

intersection F_z: $(W^{(k)z}+\Lambda^{(k)})\cap F$. Find the center $W^{(k)z+1}$ of F_z.

2. If $|W^{(k)z+1}-W^{(k)z}|<\epsilon$, ϵ *small*, then stop. Otherwise, return to step 1 with z=z+1.

For a formal development of the Algorithm of Centers, refer to Appendix A. Note that $W^{(k)}$ will always define an interior point of F as a result of the Alarm interaction technique.

5.8 CAIN Convergence

Theoretical convergence of CAIN to an efficient solution is predicated upon two concepts. The first is theoretical convergence of the Algorithm of Centers invoked at the end of each iteration. This proof is presented in Appendix A. A second consideration is convergence of DM aspiration levels toward the efficient frontier. CAIN is constructed such that at the beginning of a new iteration k+1, aspiration constraints from the previous iteration k become lower bounds for allowable objective aspirations. That is, given completion of iteration k, the aspiration constraints $A^{(k)}$ become rigid for iteration k+1. In other words, throughout iteration k+1, aspiration levels for f_i, i=1,2,...p, cannot recede below $A_i^{(k)}$, for each respective i=1,2,...,p. This mandate is insured by the Alarm interaction method (note that it is also justified since CAIN allows a DM the flexibility to modify aspiration levels <u>during a given iteration</u> as many times as desired). As a result, DM aspiration levels are forced closer and closer to the efficient frontier at each successive iteration. CAIN is therefore forced to convergence in one of two ways: 1) the Algorithm of Centers

converges to an efficient solution at the end of a given iteration and the DM chooses to accept this as the best compromise solution and terminate, or 2) the aspiration levels themselves converge to the efficient frontier. Note that method (1) will occur after only 1 iteration, but is likely to provide an imprecise solution from the standpoint of DM aspirations. Method (2) on the other hand provides a more precise approximation of DM aspirations, but will require a significantly greater number of iterations. As a possible compromise, a future computer implementation of CAIN should permit a DM to prespecify a desired number of iterations.

5.9 <u>The CAIN Algorithm</u>

5.9.1 Initialization phase (iteration k=0)

0. Map the decision space formulation into the feasible objective space: $\boldsymbol{X} \rightarrow \boldsymbol{F}$.
1. Set i=1 (objective counter).
2. Compute $I_i^{(0)}$ and $N_i^{(0)}$ for f_i. (Note: For components of R and S not yet defined (i.e. when $i < p$), their respective constraints are considered nonexistent. Mathematically,

$$\begin{aligned} r_i &\rightarrow -\infty \quad \forall i \text{ undefined} \\ s_i &\rightarrow \infty \quad \forall i \text{ undefined} \end{aligned}$$

3. Present DM with $D_i^{(0)}$ where

$$\textit{where } D_i^{(0)} = \begin{pmatrix} I_i^{(0)} \\ N_i^{(0)} \end{pmatrix}$$

Obtain absolute minimum requirement r_i where

$$N_i^{(0)} \le r_i \le I_i^{(0)}$$

Obtain satisficing threshold s_i where

$$r_i \le s_i \le I_i^{(0)} \quad \textit{if defined.}$$

4. Set i=i+1.

5. If i > p, CONTINUE TO INTERACTION PHASE. Otherwise, RETURN TO STEP 2.

5.9.2 **Interaction phase**

1. Set k=1 (iteration counter).

2. Set q=1 (ranking counter).

3. Obtain number of priority ranking groups, v, desired by DM for the objectives.

4. Obtain priority rankings P_q, q=1,2,...,v for all f_i, i=1,2,...,p from DM.

5. Set i=1 (objective function counter).

6. Check: $i \in J_q$?

   NO -- set i=i+1 and RETURN TO STEP 6.

   YES -- CONTINUE TO STEP 7.

7. Compute $I_i^{(k)}$ and $N_i^{(k)}$ for f_i.

8. Check:

$$\textit{Is } D_i^{(k)} = \begin{pmatrix} I_i^{(k)} \\ N_i^{(k)} \end{pmatrix} \textit{ an accepable range for current value of } i?$$

NO -- Eliminate $A_i^{(k)}$ $\forall i=1,2,\ldots,p$ defined for priority

q.

Check: $q = 1$?

NO -- Eliminate $A_i^{(k)}$ $\forall i=1,2,\ldots,p$ for priority

q-1. Set q=q-1. RETURN TO STEP 5.

YES -- Eliminate sets R and S. RETURN TO

INITIALIZATION PHASE, STEP 1.

YES -- Obtain DM aspiration level for f_i.

9. Set i=i+1

10. Check: $i \in J_q$?

YES -- RETURN TO STEP 7

NO -- Check: $i = p$?

NO -- RETURN TO STEP 9

YES -- Check: $q = v$?

NO -- Set q=q+1 and RETURN TO STEP 5

YES -- ITERATION COMPLETE. CONTINUE TO

CONVERGENCE PHASE.

5.9.3 Convergence Phase

1. Invoke Algorithm of Centers to converge to an efficient best compromise solution beginning with region defined by $U^{(k)} = \{ f \mid f \in F \cap (A_i^{(k)}\ \forall i=1,2,\ldots,p) \}$

   Note that at the completion of one iteration, R and S been rendered redundant and can be eliminated.

2. Present "best compromise solution" to DM. Check:

Does DM accept this solution?

YES -- TERMINATE ALGORITHM

NO -- Set k=k+1; RETURN TO INTERACTION PHASE, STEP 3

Flowchart illustrations of CAIN are given in figs. 5.3a, 5.3b, and 5.3c on the following pages.

Fig 5.3a - INITIALIZATION PHASE

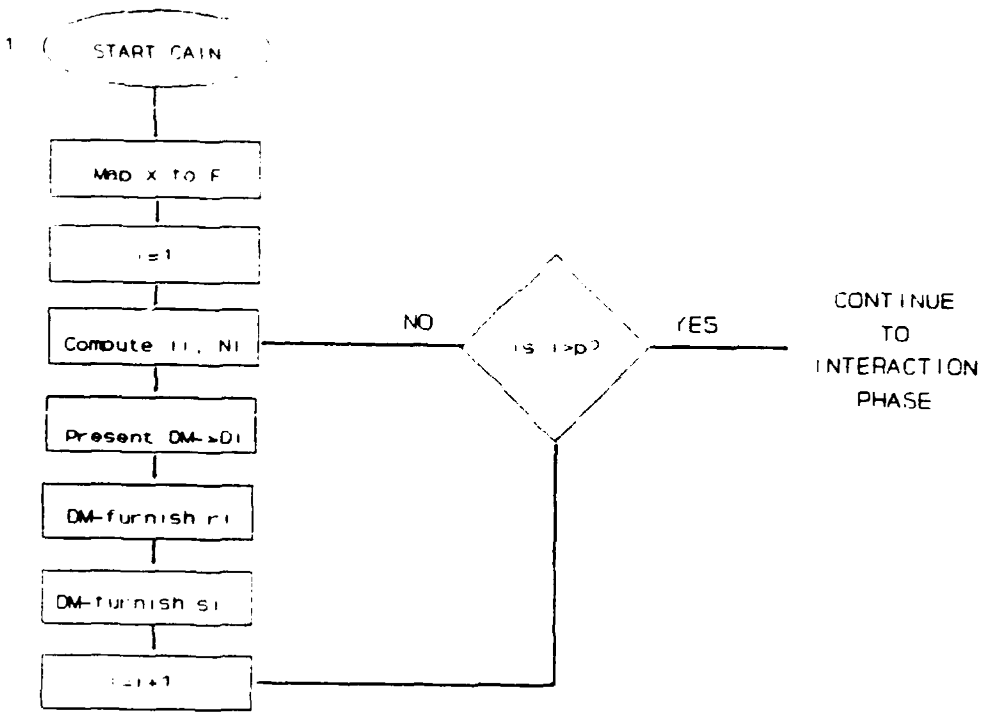


fig. 5 3b - INTERACTION PHASE

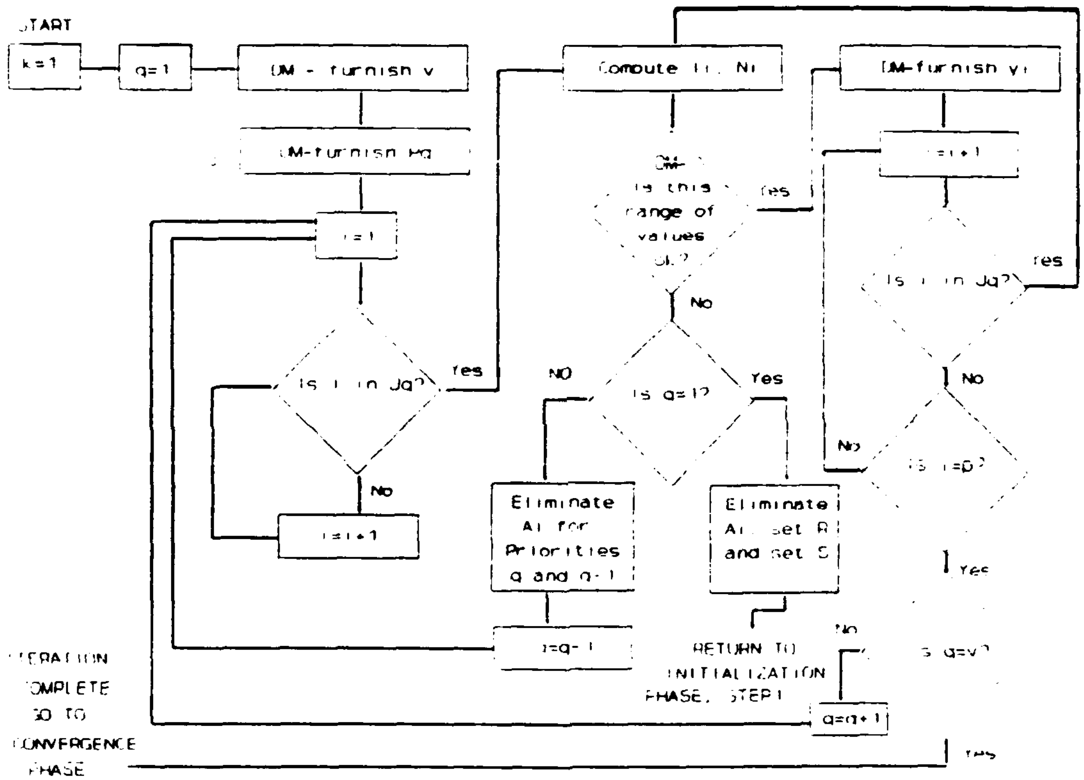
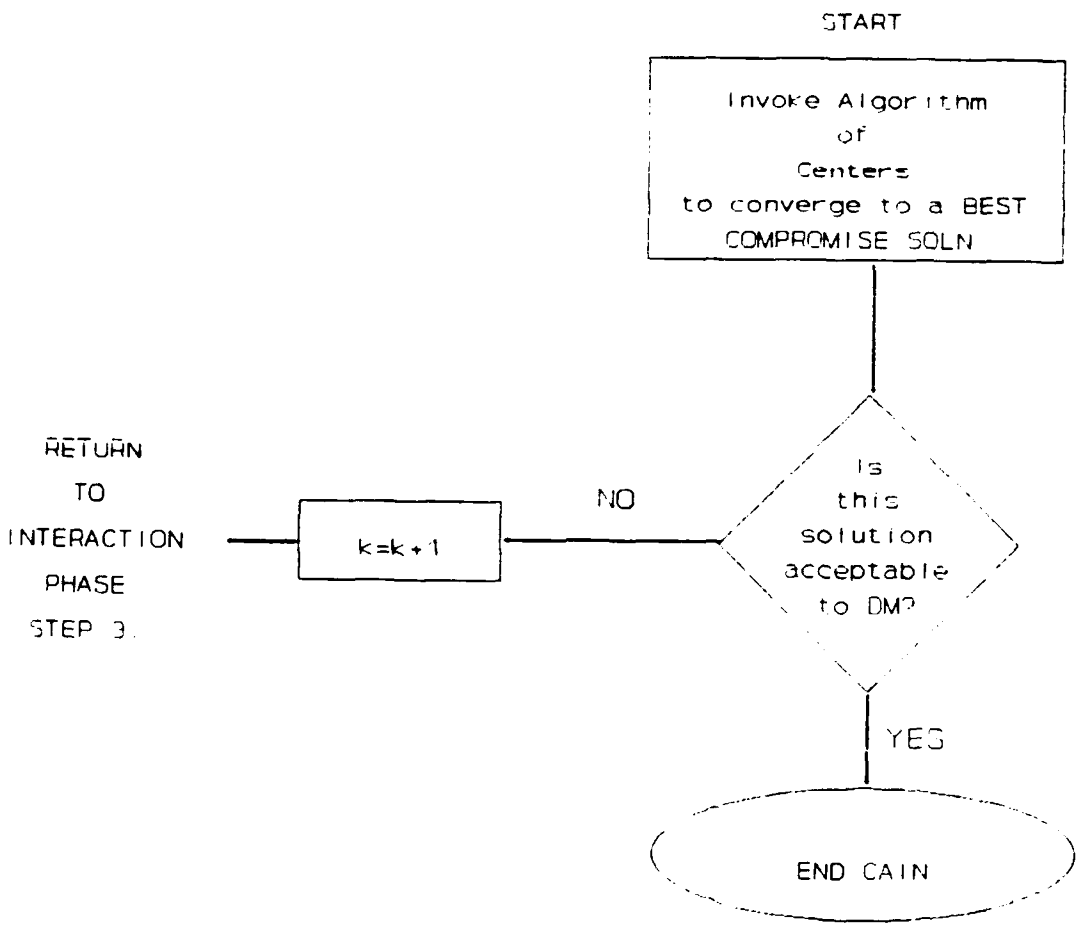


Fig 3.3c - CONVERGENCE PHASE



6.0 **CAIN EXAMPLES**

6.1 **Example 1**

In order to present both an algebraic as well as graphical illustration of CAIN, consider the two dimensional example posed in section 2.2:

$$\begin{aligned} &Max\ f_1 = x_1 + x_2 \\ &Max\ f_2 = x_3 \\ &s.t.\ \ 4x_1 + 3x_2 + 5x_3 \le 60 \\ &\qquad x_1,\ x_2,\ x_3 \ge 0 \end{aligned}$$

6.1.1 **Initialization**

Step 0: Map decision space formulation (X) into objective space formulation (F):

$$\begin{aligned} &Max\ f_1 \\ &Max\ f_2 \\ &s.t.\ \ 3f_1 + 5f_2 \le 60 \\ &\qquad f_1 \ge 0 \\ &\qquad f_2 \ge 0 \end{aligned}$$

Figure 6.1 shows the original feasible objective space (set F).

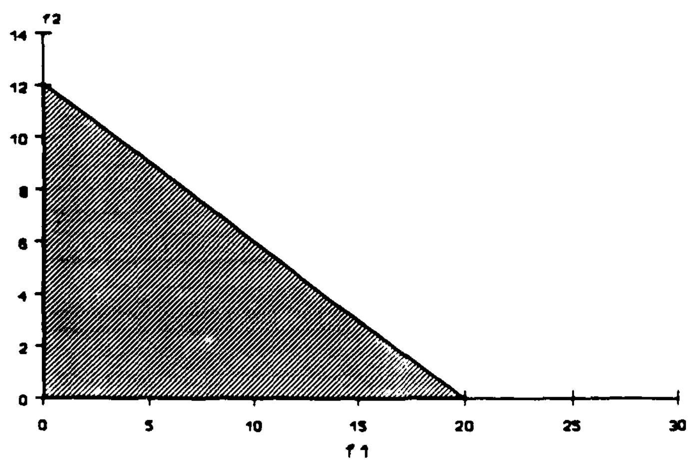


fig. 6.1

Steps 1a,2a: Set i=1. Compute ideal and nadir solutions for f_1. Recall that initially, sets R and S are nonexistent. Thus, computation of the ideal and nadir values is taken over set F.

$$I_1^{(0)} = \begin{array}{l} \text{Max } f_1 \\ s.t.\ \ 3f_1 + 5f_2 \le 60 \\ \quad\quad f_1 \ge 0 \\ \quad\quad f_2 \ge 0 \end{array} = 20; \quad N_1^{(0)} = \begin{array}{l} \text{Min } f_1 \\ s.t.\ \ 3f_1 + 5f_2 \le 60 \\ \quad\quad f_1 \ge 0 \\ \quad\quad f_2 \ge 0 \end{array} = 0$$

Step 3a: Present DM with $D_1^{(0)}$ and request absolute minimum level and any satisficing level for f_1.

$$D_1^{(0)} = \begin{pmatrix} 20 \\ 0 \end{pmatrix}$$

Assume DM furnishes the following levels:
$r_1 = 5$
$s_1 = 15$

Step 4a: i=i+1=2

Step 5a: Since $i = 2 \not> p = 2$, there must be another objective to consider. RETURN TO STEP 2.

Step 2b: Based upon DM inputs for f_1, compute ideal and nadir values for f_2 as follows. Note that sets R and S are no longer nonexistent and that redundant constraint $f_1 \ge 0$ can be eliminated.

$$I_2^{(0)} = \begin{array}{l} \text{Max } f_2 \\ s.t.\ \ 3f_1 + 5f_2 \le 60 \\ \quad\quad f_1 \ge 5 \\ \quad\quad f_1 \le 15 \\ \quad\quad f_2 \ge 0 \end{array} = 9; \quad N_2^{(0)} = \begin{array}{l} \text{Min } f_2 \\ s.t.\ \ 3f_1 + 5f_2 \le 60 \\ \quad\quad f_1 \ge 5 \\ \quad\quad f_1 \le 15 \\ \quad\quad f_2 \ge 0 \end{array} = 3$$

Present DM with $D_2^{(0)}$ and request input of absolute minimum requirement and any satisficing level for f_2:

$$D_2^{(0)} = \begin{pmatrix} 9 \\ 3 \end{pmatrix}$$

Assume DM furnishes the following levels:

$r_2 = 4$

s_2 not defined

Step 4b: i=i+1=3

Step 5b: Since i=3 > p=2, initialization is complete. CONTINUE TO INTERACTION PHASE.

Thus, sets R and S are now defined as:

$$R = \begin{Bmatrix} f_1 \geq 5 \\ f_2 \geq 4 \end{Bmatrix}; \quad S = \begin{Bmatrix} f_1 \leq 15 \\ f_2 \leq \infty \end{Bmatrix}$$

Note that set S immediately becomes redundant due to the continuous updating of ideal and nadir values during the Alarm interaction segment. The reduced feasible objective space prior to beginning the interaction phase is shown in fig 6.2.

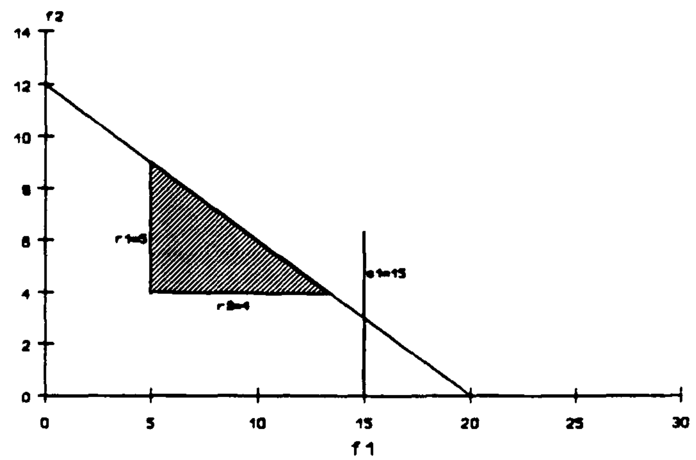


fig. 6.2

6.1.2 **Interaction phase**

Steps 1,2: $k=1$; $q=1$

Step 3a: Obtain number of priority ranking groups desired by DM for the objectives.

> **Assume DM desires to rank the objectives individually:**
> **v=2**

Step 4a: Obtain DM priority rankings of objectives:

> **Assume DM ranks objectives in order of relative importance as:**
> **1) f_2**
> **2) f_1**

Step 5a: Set $i=1$

Step 6a: Checks and counter updates:

$$i = 1 \in J_1? \rightarrow no \rightarrow i=i+1=2 \in J_1? \rightarrow yes$$

Step 7a: CAIN recognizes the highest priority objective from Step 6a. Obtain ideal and nadir solutions for f_2 :

$$I_2^{(1)} = \begin{array}{l} \text{Max } f_2 \\ s.t. \;\; 3f_1 + 5f_2 \leq 60 \\ \qquad f_1 \geq 5 \\ \qquad f_2 \geq 4 \end{array} = 9 \; \ldots \; N_2^{(1)} = \begin{array}{l} \text{Min } f_2 \\ s.t. \;\; 3f_1 + 5f_2 \leq 60 \\ \qquad f_1 \geq 5 \\ \qquad f_2 \geq 4 \end{array} = 4$$

Step 8a: Present DM with $D_2^{(1)}$. If DM agrees that this is an acceptable range for f_2, then obtain aspiration level for f_2.

$$D_2^{(1)} = \begin{pmatrix} 9 \\ 4 \end{pmatrix}$$

> **Assume DM concurs with the allowable range and inputs the following aspiration level for f_2:**
>
> $$y_2 = 5$$

Step 9a: i=i+1=3

Step 10a: Check and counter updates:

$$i = 3 \in J_1? \rightarrow no \rightarrow i=3 > p=2? \rightarrow yes \rightarrow q=1=s=2? \rightarrow no \rightarrow q=q-1=2$$

CAIN recognizes no more objectives as priority 1 and that a priority 2 objective exist. ALGORITHM RETURNS TO STEP 5 with q=2.

Step 5b: Set i=1.

Step 6b: Checks and counter updates:

$$i = 1 \in J_2? \rightarrow yes \rightarrow GO\ TO\ STEP\ 7$$

Step 7b: Based upon DM inputs to this point, ideal and nadir values for priority 2 objective f_1 are computed:

$$I_1^{(1)} = \begin{array}{l} \text{Max } f_1 \\ \text{s.t. } 3f_1 + 5f_2 \le 60 \\ \quad f_1 \ge 5 \\ \quad f_2 \ge 5 \end{array} = 11.67 \ \ldots \ N_1^{(1)} = \begin{array}{l} \text{Min } f_1 \\ \text{s.t. } 3f_1 + 5f_2 \le 60 \\ \quad f_1 \ge 5 \\ \quad f_2 \ge 5 \end{array} = 5$$

Step 8b: Present DM with $D_1^{(1)}$. If this range for f_1 is acceptable, obtain aspiration level for f_1.

$$D_1^{(1)} = \begin{pmatrix} 11.67 \\ 5 \end{pmatrix}$$

Assume DM concurs with allowable range and furnishes the following aspiration level for f_1:

$$y_1 = 7$$

Step 9b: i=i+1=2

Step 10b: Updating counters as given in the algorithm, CAIN branches to the Convergence phase. That is, CAIN recognizes that input requirements have been met for a complete iteration since there are no remaining objectives or priorities. Based upon DM inputs, the reduced feasible objective space after iteration 1 is shown in fig. 6.3.

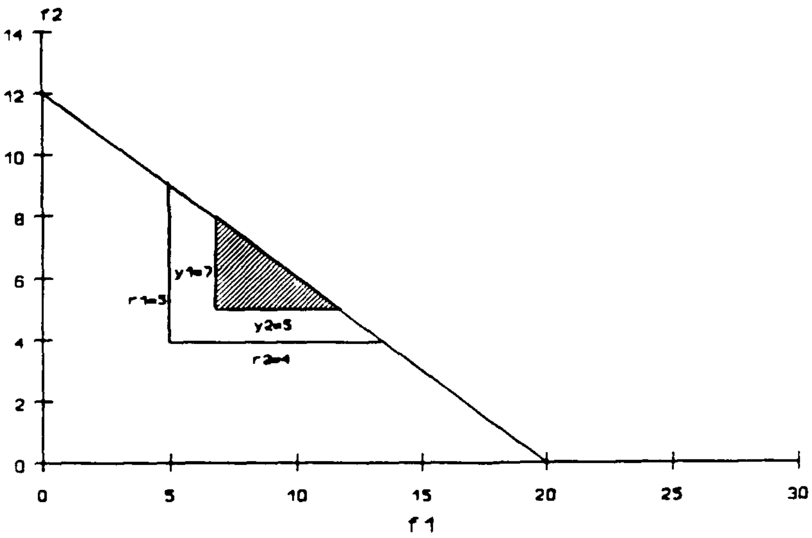


fig. 6.3

6.1.3 Convergence Phase

Step 1: CAIN invokes the Algorithm of Centers to converge to a best compromise solution beginning with region $U^{(1)} = F \cap A^{(1)}$

This is illustrated in fig. 6.4.

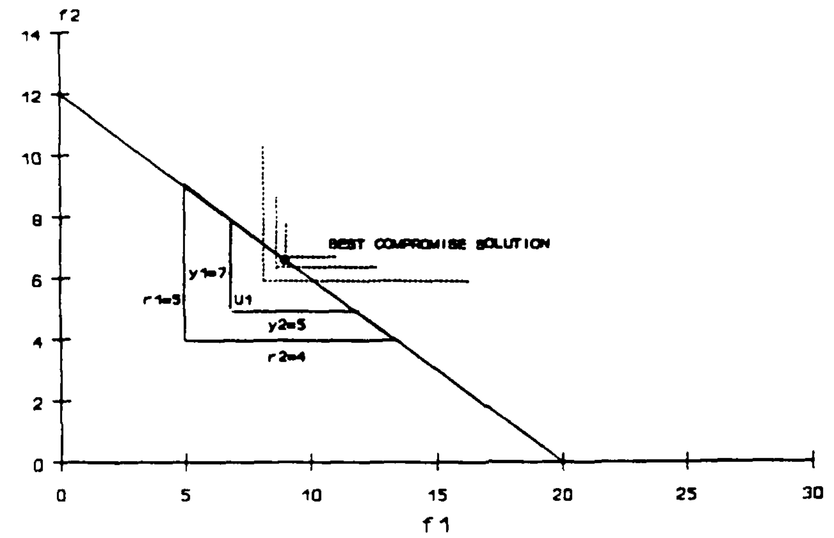


fig. 6.4

Step 2: DM can choose to accept this best compromise solution, or continue narrowing the feasible space with tighter aspiration levels. Assuming DM is satisfied, the algorithm terminates at the best compromise solution of fig. 6.4.

6.2 **Example 2**

As a demonstration of additional CAIN features, reconsider Example 1. Assume that at Step 8b, a DM is not satisfied with the range of values presented by $D_1^{(1)}$:

$$D_1^{(1)} = \begin{pmatrix} 11.67 \\ 5 \end{pmatrix}$$

Step 8b: Recall that q=2 and since the DM does not concur with $D_1^{(1)}$, $A_1^{(1)}$ is not yet defined. Therefore, there are no priority 2 aspiration constraints to eliminate (f_1 is the only 2nd priority objective). Since $q > 1$, $A_2^{(1)}$, the priority 1 constraint is eliminated and the priority counter q is reset to q-1=1. CAIN

then returns to Step 5 with the feasible objective space defined as in fig 5.2.

<u>Steps 5c, 6c, 7c, 8c</u>: After updating the counters (q=1, i=1), the algorithm requests the DM refurnish aspirations for f_2 based upon $D_2^{(1)}$ from Step 8a. Note the desired aspiration should be lowered in order to broaden the range for $D_1^{(1)}$.

$$D_2^{(1)} = \begin{pmatrix} 9 \\ 4 \end{pmatrix}$$

Assume DM refurnishes aspiration for f_2 as follows:

$$y_2 = 4.5$$

<u>Steps 9c, 10c</u>: Counters are updated (q=2) and CAIN returns to Step 5 for recomputation of $D_1^{(1)}$ based upon the new aspiration for f_2.

<u>Steps 5d, 6d</u>: Update counters → i=1.

<u>Step 7d</u>: Recomputation of $I_1^{(1)}$ and $N_1^{(1)}$:

$$I_1^{(1)} = \begin{array}{l} \quad Max\ f_1 \\ s.t.\ \ 3f_1 + 5f_2 \le 60 \\ \qquad f_1 \ge 5 \\ \qquad f_2 \ge 4.5 \end{array} = 13.5 \ \ldots \ N_1^{(1)} = \begin{array}{l} \quad Min\ f_1 \\ s.t.\ \ 3f_1 + 5f_2 \le 60 \\ \qquad f_1 \ge 5 \\ \qquad f_2 \ge 4.5 \end{array} = 5$$

Thus, the achievable ideal for f_1 has increased from 11.67 to 13.5.

<u>Step 8d</u>: Present DM with updated $D_1^{(1)}$:

$$D_1^{(1)} = \begin{pmatrix} 13.5 \\ 5 \end{pmatrix}$$

If the DM concurs with this range, a new aspiration is requested for f_1.

Assume DM concurs and furnishes aspiration for f_1 as follows:

$$y_1 = 12$$

Therefore, the iteration is completed and CAIN branches to the convergence phase beginning with the region shown in fig 6.5

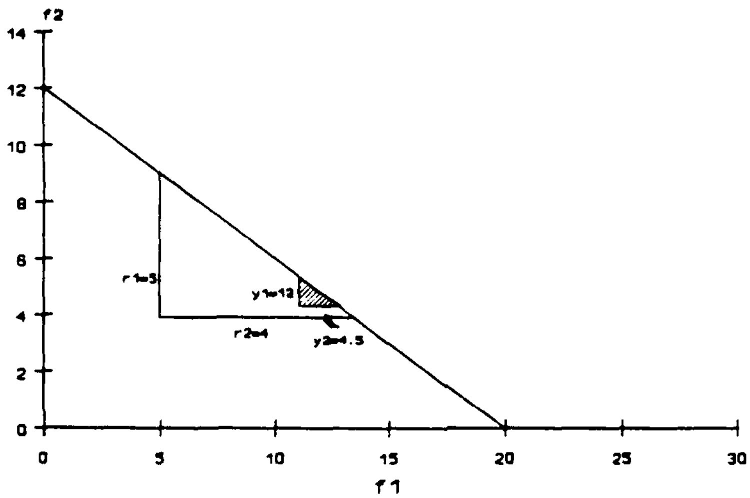


fig. 6.5

Recall that once an iteration is complete, the defined aspiration levels for that iteration become fixed. Aspirations are only permitted readjustment <u>during</u> an iteration in order to insure convergence.

7.0 SUBJECTIVE COMPARISONS

7.1 CAIN vs. Traditional methods

The traditional MOLP solution strategies of section 3 suffer from one of two glaring deficiencies. In general, these techniques are either 1) restricted to the set of efficient extreme points as a result of implementing the simplex algorithm as an efficient solution generator, or 2) too computationally complex to be of any real practical value. As a rule, traditional techniques employing a prior articulation or progressive articulation of DM preferences suffer from the first inadequacy (see sections 3.2 and 3.4). On the other hand, computational complexity and lack of practicality are common characteristics of the posteriori approaches as they attempt to construct the entire efficient frontier (see section 3.3.2). In addition to these inadequacies, the traditional progressive strategies nearly all require DM interactions involving comparisons of multiple, often similar alternatives (sections 3.4.1 - 3.4.3) or burdensome tradeoff inquiries (section 3.4.4). Traditional aspiration based approaches also tend to suffer from the simplex restrictions (section 3.5.1).

One particular methodology, the method of Wierzbicki (section 3.5.2) has shown promise in alleviating the extreme point restrictions. As a result, this approach has been employed as the efficient alternative generator in some of the newer interactive methodologies. Unfortunately, the Wierzbicki strategy assumes DM ability to furnish aspiration levels "near" the efficient frontier in order to be effective.

CAIN has attempted to improve on all of these potential deficiencies. By not relying on simplex techniques for efficient alternative generation, it is not bound to nondominated extreme points. Furthermore, CAIN interaction requires nothing more than levels of aspiration. This simple progressive articulation is further enhanced by the Alarm interaction technique. As a result of requiring no alternative comparisons or tradeoff inquiries, the potential for DM inconsistency is eliminated. Finally, although computationally complex, the Algorithm of Centers is conceptually comprehendible for the average DM (unlike construction of the efficient frontier discussed in section 3.3.2) and does not assume any prior understanding of efficiency.

7.2 CAIN vs. New (Tchebycheff) methods

While obviously superior to most traditional methods, a subjective comparison of CAIN to more modern MOLP approaches should reveal a more realistic measure of its standing. Generally, these new approaches use variations of Wierzbicki (Tchebycheff) weighting techniques for generating efficient solutions. They therefore, like CAIN, are not bound to extreme points as are many of the traditional methodologies. Since both CAIN and these modern strategies are capable of achieving any alternative on the efficient frontier, the main attribute of comparison becomes the various interaction techniques employed.

While CASE, KD, and SC (see sections 4.2 - 4.4) are all able to explore the entire efficient frontier, they also require a DM to make some form of comparisons between alternatives. If a DM is

unable to consistently perform this task, algorithm convergence is immediately placed in jeopardy. As alluded to above, CAIN requires no comparison of alternatives. Although a DM is permitted to rank order the objectives during the CAIN interaction segment, this information is not mandatory. CAIN guarantees convergence using only DM furnished aspirations guided by the Alarm interaction system. Furthermore, as a result of Alarm, the Algorithm of Center will always converge to an efficient solution at least as good as DM aspirations. On the other hand, the other algorithms permit input of infeasible aspirations. It is then left to the Tchebycheff approach to determine which objectives are compromised to project back to the efficient frontier.

Subjectively, CAIN appears to possess some advantages over other comparable MOLP strategies. This is particularly true in the areas of interaction and convergence. Nevertheless, since CAIN is still in its infant development stages, actual experimental comparisons of CAIN vs. alternative approaches must be accomplished before any concrete conclusions can be drawn.

8.0 CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER RESEARCH

This report has described the development of a new aspiration based interior point algorithm for solving the multiple objective linear programming (MOLP) problem. The method, known as CAIN (Convergent Aspiration based INterior method), guides a decision maker (DM) to a best compromise solution through implementation of a new DM interaction technique. This technique, called "ALaRM" (Aspiration Level Range Method) assists in holding DM aspiration levels feasible at all times while concurrently keeping a DM abreast of achievable objective values. Once DM aspiration levels are furnished, CAIN converges to an efficient best compromise solution (based upon the defined aspirations) via the Algorithm of Centers, an interior point optimization approach for Multiple Criteria Decision Making (MCDM).

While subjectively, CAIN appears to possess advantages over other aspiration techniques for MOLP (section 7), the computational complexity of the Algorithm of Centers requires computer implementation to be practical. Thus, the next logical step is development of a computer algorithm for CAIN. Furthermore, as of this writing, research is being conducted to improve the Algorithm of Centers (Haas, 1990). The research is geared toward reducing the number of iterations, and thus enhancing its speed of convergence. This research should be closely monitored for any features which might benefit CAIN.

Ideally, development of CAIN has marked the beginning of a new

practical MOLP solution technique. At the very least, it is hoped that additional interest has been kindled for further research into the applicability of interior point solution techniques to multiple criteria decision problems.

REFERENCES

1. Abhyankar, T., Morin T., and Trafalis, T. *Efficient Faces of Polytopes: Interior Point Algorithms, Parameterization of Algebraic Varieties, and Multiple Objective Optimization*, Contemporary Mathematics 114 (1990), 319-341.

2. Arthur, J. and Ravindran A. *An Efficient Goal Programming Algorithm Using Constraint Partitioning and Variable Elimination*, Management Science 24(8) (1978), 867-868.

3. Bayer, D. and Lagarias, J. *The Nonlinear Geometry of Linear Programming*, I,II, Trans. Amer. Math. Soc. 314 (1989), 499-526; 527-581.

4. Bazaraa, M., Jarvis, J. and Sherali, H. Linear Programming and Network Flows, John Wiley and Sons, New York (1990).

5. Benayoun, R,. et. al. *Linear Programming with Multiple Objective Function: Step Method (STEM)*, Mathematical Programming 1 (1971), 366-375.

6. Chankong, V. and Haimes, Y. Multiobjective Decision Making: Theory and Methodology, North Holland Publishing Co., New York (1983).

7. Charnes, A. and Cooper, W. Management Models and Industrial Applications of Linear Programming, John Wiley and Sons, New York (1961).

8. Dell R., and Karwan, M. *An Interactive MCDM Weight Space Reduction Method Utilizing a Tchebycheff Utility Function*, to appear in Naval Research Logistics, Working Paper, Department of Industrial Engineering, SUNY, Buffalo (1989).

9. Dyer, J. and Sarin R. Multicriteria Decision Making, Prepared for the Encyclopedia of Computer Science and Technology, Marcel Dekker, New York (1979).

10. Dyson, R. *Maximin Programming, Fuzzy Linear Programming, and Multi-Criteria Decision Making*, J. Oper. Res. Soc., (1980), 263-267.

11. Ecker, J. and Kouada, I. *Finding Efficient Points for Linear Multiple Objective Programs*, Mathematical Programming 8 (1975), 375-377.

12. Ecker, J. and Kouada, I. *Finding All Efficient Extreme Points for Linear Multiple Objective Programs*, Mathematical Programming 14 (1978), 249-261.

13. Ecker, J., et. al. *Generating All Maximal Efficient Faces for Multiple Objective Linear Programs*, Journal of Optimization Theory and Application, 30 (1980), 353-381.

14. Evans, J. and Steuer, R. *Generating Efficient Extreme Points in Linear Multiple Objective Programming: Two Algorithms and Computing Experience*, in Cochrane and Zeleny, Multiple Criteria Decision Making, University of South Carolina Press (1973).

15. Farquhar, P. *A Survey of Multiattribute Utility Theory and Applications*, in M.K. Starr and M. Zeleny (Eds.) Multiple Criteria Decision Making, TIMS Studies in the Management Sciences 6 (1977), 59-89.

16. Fichefet, J. *GPSTEM: An Interactive Multiobjective Optimization Method*, in A. Prekopa, Progress in Operations Research, vol. 1, North-Holland, Amsterdam (1979).

17. Gal, T. *A General Method for Determining the Set of All Efficient Solutions to a Linear Vector Maximum Problem*, Report No. 76/12, Institut fur Wirtschaftswissenschaften, Aachen, Germany (1976).

18. Haas, A. "Method of Centers", M.S. Thesis, Purdue University (1991).

19. Hazen, G. and Morin T. *Steepest Ascent Algorithms for Nonconical Multiple Objective Programming*, J. Math. Anal. Appl. 100 (1984), 188-221.

20. Huard, P. *Resolution of Mathematical Programming with Nonlinear Constraints by the Method of Centers*, in Abadie, J., Nonlinear Programming, North-Holland, Amsterdam (1967).

21. Ijiri, Y. Management Goals and Accounting for Control, North-Holland, Amsterdam (1965).

22. Isermann, M. *The Enumeration of the Set of All Efficient Solutions for a Linear Multiple Objective Program*, Operational Research Quarterly 28 (1977), 711-725.

23. Isermann, M. *The Enumeration of All Efficient Solutions for a Linear Multiple-Objective Transportation Problem*, Naval Research Logistics Quarterly 26 (1979), 123-139.

24. Khachian, L. *A Polynomial Algorithm for Linear Programming*, Dokl. Akad. Nauk SSSR 244 (1979), 1093-1096.

25. Karmarkar, N. *A New Polynomial-Time Algorithm for Linear Programming*, Combinatorica 4 (1984), 373-395.

26. Keeney, R. and Raiffa, H. Decisions with Multiple Objectives: Preferences and Value Tradeoffs, John Wiley and Sons, Inc. New York (1976).

27. Klee, V. and Minty G. *How Good is the Simplex Method*, in Shisha, O., Inequalities III, Academic Press, New York (1972).

28. Kornbluth, J. *A Survey of Goal Programming*, OMEGA 1 (1973), 193-205.

29. Lee S. *Goal Programming for Decision Analysis of Multiple Objectives*, Sloan Management Review 14 (1973), 11-24.

30. Lofti, V., Stewart, T., and Zionts, S. *An Aspiration-Level Interactive Method for Multiple Criteria Decision Making*, Working Paper #711, School of Management, SUNY, Buffalo (1989).

31. Malakooti, B. and Ravindran, A. *Experiments with an Interactive Paired Comparison Simplex Method for MOLP Problems*, Annals of Operations Research 5 (1985/1986), 575-597.

32. Marsten R. et. al. *Implementation of a Dual Affine Interior Point Algorithm for Linear Programming*, Working Paper, University of Arizona, Tuscon, AZ (1988).

33. McShane, K. et. al. *An Implementation of a Primal-Dual Interior Point Method for Linear Programming*, Working Paper (1988).

34. Morin, T. and Trafalis, T. *A Polynomial-Time Algorithm for Finding an Efficient Face of a Polyhedron*, Technical Report, Purdue University, West Lafayette, Ind. (1989).

35. Philip, J. *An Algorithm for the Vector Maximization Problem*, Mathematical Programming 2 (1972), 207-209.

36. Ravindran, A. *Tutorial On Goal Programming*, Presented at the Spring Annual Conference of IIE (1985).

37. Renegar, J. *A Polynomial-time Algorithm, based on Newton's Method, for Linear Programming*, Mathematical Programming 40 (1988), 59-93.

38. Roy, B. *Classement et Choix en Presence de Points de Vue Multiples (La Methode ELECTRE)*, Revue d'Informatique et de Recherch Operationelle 8 (1968), 57-75.

39. Sage, A. Methodology for Large-Scale Systems, McGraw-Hill, New York (1977).

40. Steuer, R. *A Five Phase Procedure for Implementing a Vector-Maximum Algorithm for Multiple Objective Linear Programming Problems*, in H. Thiriez and S. Zionts (Eds.), Multiple Criteria Decision Making, Springer-Verlag, Berlin (1976a).

41. Steuer, R. *Multiple Objective Linear Programming with Interval Criterion Weights*, Management Science 23 (1976b), 305-316.

42. Steuer, R. and Choo, E. *An Interactive Weighted Tchebycheff Procedure for Multiple Objective Programming*, Mathematical Programming 26 (1983), 326-344.

43. Trafalis, T. *Efficient Faces of a Polytope: Interior Methods in Multiple Objective Optimization*, Ph.D. dissertation, Purdue University (1989).

44. Vaidya, R. *A Locally Well-Behaved Potential Function and a Simple Newton-Type Method for Finding the Center of a Polytope*, Technical Report, AT&T Bell Laboratories, Murray Hill, N.J. (1987).

45. Wierzbicki, A. *The Use of Reference Objectives in Multiobjective Optimization*, Working Paper, International Institute for Applied Systems Analysis, Laxenburg, Austria (1979).

46. Yoon, Y., Lofti, V. and Zionts, S. *Convergent Aspiration-Level Search (CASE) Method for Multiple Objective Linear Programming Problem: Theory and Comparative Tests*, Working Paper (1991).

47. Yu, P. and Zeleny, M. *The Set of all Nondominated Solutions in Linear Cases and a Multicriteria Simplex Method*, J. Math. Anal. Appl. 49 (1975), 430-468.

48. Yu, P. Multiple-Criteria Decision Making, Plenum Press, New York (1985).

49. Zeleny, M. Linear Multiobjective Programming, Springer, Berlin/Heidelberg (1974a).

50. Zeleny, M. *A Concept of Compromise Solutions and the Method of the Displaced Ideal*, Computers and Operations Research 1(4) (1974b), 479-476.

51. Zeleny, M. Multiple Criteria Decision Making, McGraw-Hill, New York (1982).

52. Zionts, S. and Wallenius, J. *An Interactive Programming Method for Solving the Multiple Criteria Problem,* Management Science 22(6) (1976), 652-663.

A.0 APPENDIX A - ALGORITHM OF CENTERS

A.1 Interior Point Approach for MOLP (Trafalis, 1989)

This section begins by first presenting some necessary notation and a problem statement.

Let P be a full-dimensional bounded polytope in $\mathbf{R}^n$ described by linear inequalities:

$$P = \bigcap_{i=1}^{m} H_i, \text{ where } H_i = \{ x \mid a_i^T x \geq b_i \},$$

and $\mathbf{R}^n$ is ordered by a constant closed polyhedral cone Λ. Recall that a set Λ is a cone if $ay \in \Lambda$ for any a > 0 and $y \in \Lambda$. A polyhedral cone is a cone that is also a polyhedron. Thus if Λ is also a polyhedral cone, then it can be represented by:

$$\Lambda = \{ x \mid \Lambda x \geq 0 \},$$

where Λ is a matrix of proper dimension. Hereafter, the same notation shall be used for both the polyhedral cone and its defining matrix.

Consider the following problem:

PROBLEM: *Find the set of points* $\overline{P}$ *where:*

$$\overline{P} = \{ \overline{x} : (\overline{x} + \Lambda) \cap P = \{ \overline{x} \}\}.$$

This set is called the *efficient frontier* of P and is a subset of the boundary of P. Note that the term efficient is used instead of maximal in order to avoid confusion with the definition of a facet as a maximal face with respect to set inclusion. Also note that faces other than n-1 -dimensional faces (i.e., facets) can be

efficient.

Definition A.1.1 (Yu, 1985): *A face F of P is called efficient iff F is a subset of the efficient frontier of P.*

Remark: The efficient frontier of P is the union of all efficient faces of P.

This appendix shall be concerned with finding a point on $\overline{P}$. The concept of an analytical center of a bounded polyhedron (Sonnevend, 1985 and Bayer and Lagarias, 1986) is fundamental to the approach.

A.2 Algorithm of Centers for finding an efficient face

This section describes the method of centers (Huard, 1967) applied to vector optimization. A similar approach was followed by Renegar (1986) in the single objective linear case.

Let x^k be an interior point of P and consider the intersection P_k, of $x^k + \Lambda$ and P. Next, find the center x^{k+1} of P_k and start again with x^{k+1} instead of x^k. That is, take a sequence of points $\{x^k\}$ that is in general infinite. Section A.3 shall demonstrate that this sequence converges to an efficient solution. In formal notation, the Algorithm of Centers is presented as follows:

Algorithm of Centers

0. Set k=0.

1. Let x^k be an interior point of P. Consider the intersection P_k of $x^k + \Lambda$ and P. Find the center x^{k+1} of

P_k.

2. If $| x^{k+1} - x^k | < \epsilon$, then stop. Else return to 1 with $k \leftarrow k+1$.

The above algorithm has the following characteristics:

- It is an iterative procedure and the round-off errors do not accumulate.
- It approaches an efficient solution from below through strictly increasing values (with respect to the cone ordering).
- It is an interior point algorithm whose path is not influenced by the peculiarities of the feasible region.

Normally, in order to find the analytical center of a convex polytope defined by linear inequalities, a nonlinear program must be solved. Recently, several researchers realized the usefulness of the center of a convex set, and especially the center of a polyhedron (Bayer and Lagarias, 1986, and Sonnevend, 1985). It can be shown that if a polyhedron is defined by a set of linear inequalities, then methods of computing its center exist. (Trafalis, 1989).

Next, the concept of a potential function for the set P_k is described. Let P be the given polyhedron (which is assumed full dimensional), Λ be the ordering cone, and x^k be a point in *int*(P). Define:

$$P_k = \{ x^k + \Lambda \} \cap P,$$

where

$$P = \bigcap_{i=1}^{m} H_i,$$

$$H_i = \{ x \mid a_i^T x \geq b_i \},$$

and

$$\Lambda = \{ x \mid \Lambda_s^T x \geq 0, \; s=1,2,\ldots,p \}.$$

It follows at once that P_k is a bounded polytope. Next define the potential function:

$$f_{P_k} : int(P_k) \to R,$$

such that

$$f_{P_k}(x) = \sum_{i=1}^{m} log(a_i^T x - b_i) + \sum_{s=1}^{p} log(\Lambda_s^T x - \Lambda_s^T x^k).$$

The objective is to maximize f_{P_k}. It can easily be proven that f_{P_k} is strictly concave on $int(P_k)$ (Bayer and Lagarias, 1986). Thus, it has a unique maximizer x_{k+1}, and x_{k+1} is a solution of

$$df_{P_k}(x) = 0,$$

or

$$\sum_{i=1}^{m} \frac{a_i^T}{a_i^T x - b_i} + \sum_{s=1}^{p} \frac{\Lambda_s^T}{\Lambda_s^T x - \Lambda_s x^k} = 0.$$

This is a nonlinear system of equations that can be solved by Newton's method (Sonnevend, 1985, and Vaidya, 1987).

A.3 Convergence of the Algorithm

Proof of convergence of the Algorithm of Centers is now presented using the following notation:

$$\Lambda(\lambda) = \{ x \mid \Lambda x \geq \Lambda\lambda \},$$

$$P(\lambda) = P \cap \Lambda(\lambda),$$

$$S(\lambda) = P \cap int(\Lambda(\lambda)),$$

$$F(\lambda) = bdr(P(\lambda)),$$

$$I(\lambda) = int(P(\lambda)),$$

$$P_k = P(x^k), \text{and}$$

$$S_k = S(x^k).$$

The following observations will be helpful for the convergence proofs:

If $S(\lambda) \neq \emptyset$, *then* $I(\lambda) \neq \emptyset$, *and*

If $I(\lambda) = \emptyset$ *then* $S(\lambda) = \emptyset$.

Finally, the following convergence theorem for multiple objective optimization shall be used:

Theorem A.3.1 (Hazen and Morin, 1984). Let $<$ and $\leq$ be binary relations on X such that:

(a) $x < y$ iff $x \leq y$ *and not* $y \leq x$,

(b) $<$ is transitive, and

(c) the sets $x \leq$ *and* $\leq x$ *are closed*.

Let T *be a point-to-set map on* X, *and let* Ω *be a desirable*

set such that:

(d) If $x \in \Omega$ *and* $y \in Tx$, *then* $x<y$, *and*

(e) Ω is closed from below under $\leq$ *on* $\{X - \Omega\}$.

Suppose a sequence $\{x^k\}$ *is generated satisfying* $x^{k+1} \in Tx^k$ *and no member of* $\{x^k\}$ *is in* Ω. *If* $x^k \subseteq S$ *for some compact set* $S \subseteq X$, *then every limit point of* $\{x^k\}$ *is in* Ω.

Using this, the following proof can be constructed.

Theorem A.3.2 (Trafalis, 1989): *The sequence* $\{x^k\}$ *constructed by the Algorithm of Centers is finite and either the last element is a weakly efficient point, or else every subsequence of* x^k *converges to an efficient point.*

Proof: Invoke Theorem A.3.1 with the ordering in $\mathbf{R}^n$ defined by the cone Λ. That is,

$$x \geq y \text{ iff } x-y \in \Lambda, \text{ and}$$

$$x > y \text{ iff } x-y \in \bar{\Lambda} = \Lambda - \{0\}.$$

It is easy to check (a), (b), and (c) of the general convergence theorem. Next, define a point-to-set map on $\mathbf{R}^n$ as follows:

$$T : R^n \to 2^{R^n},\ Tx = \{x + \bar{\Lambda}\}.$$

Define the set of desirable points to be:

$$\Omega = \{\bar{y} \in P : \{\bar{y} + \Lambda\} \cap P = \{\bar{y}\}\}.$$

Condition (d) of Theorem A.3.1 follows at once from the fact that if $x \notin \Omega$ *and* $y \in Tx$ *, then* $\{y-x\} \in \bar{\Lambda}$. Next, it will be shown that T is closed form below under $\leq$ *on* $\{P-\Omega\}$. Let $\{x^k, y^k\}$ be such that:

$$x^k \to x,\ y^k \in T(x^k) \text{ and } y^k \to y.$$

Taking the limit, $\{y-x\} \in \bar{\Lambda}$, *i.e.* $y \in T(x)$. If $S_k = \emptyset$, then $P \cap int(\Lambda(x^k)) = \emptyset$. Hence, either x is a weakly efficient solution or by the above observation, $I(x^k) \neq \emptyset$. In the latter case, the following nonlinear programming problem must be solved:

$$\begin{gathered} \max f_{P_k}(x) \\ s.t.\ x \in I(x^k). \end{gathered}$$

Note that this problem has a unique solution since P_k is bounded, the sequence of points x^k has the property that no member of it is in Ω, and P is compact. Therefore, invoking Theorem A.3.1 yields the desired result, namely that the limit of every subsequence is an efficient point.

Q.E.D.

The main computational burden of the Algorithm of Centers is related to the determination of the analytical centers of different iterations. A Newton's path following technique can be used for this algorithm and can be shown to be polynomial in the number of iterations. For additional details, see Trafalis (1989).